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

*by Nathan Ensmenger**

ABSTRACT

Over the course of the 1960s and 1970s, male computer experts were able to successfully transform the "routine and mechanical" (and therefore feminized) activity of computer programming into a highly valued, well-paying, and professionally respectable discipline. They did so by constructing for themselves a distinctively masculine identity in which individual artistic genius, personal eccentricity, anti-authoritarian behavior, and a characteristic "dislike of activities involving human interaction" were mobilized as sources of personal and professional authority. This article explores the history of masculine culture and practices in computer programming, with a particular focus on the role of university computer centers as key sites of cultural formation and dissemination.

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 to be 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 over 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 other-

---

* School of Informatics and Computing, Indiana University, Bloomington, IN 47408; nensmeng@indiana.edu.

[1] Joseph Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (San Francisco, 1976), 116.

wise 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 interludes 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, or engineers building useful products to benefit society, but mere junkies in search of a fix. That such myopic and socially maladjusted tinkerers were being accorded such a prominent and influential role in the construction of the essential structures of the modern information society was, for Weizenbaum, the dangerous and disturbing consequence of a reckless 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 that Weizenbaum was deriding the obsessive tendencies of the computer bum, a powerful counternarrative was emerging in which such single-minded focus was lauded as desirable, possibly even heroic. In this 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 as "good news, maybe the best since psychedelics." Where Weizenbaum perceived in computerization the realization of impersonal, bureaucratic and authoritarian imperatives, Brand saw revolutionary potential and the empowerment of individuals. Via the computer, revolutionary citizens/programmers could appropriate Cold War technologies for the purposes of progressive social transformation.[4]

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

---

[2] Ibid.

[3] Sherry Turkle, *The Second Self: Computers and the Human Spirit* (New York, 1984).

[4] Fred Turner, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism* (Chicago, 2006), 116–7.

cathode ray tube display, Brand was clearly less interested in Spacewar than he was in its computer "hacker" developers.[5] While he acknowledged the double-edged bite of the "hacker" epithet (which he deemed both "a term of derision and the ultimate compliment"), Brand's 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 the degree that they violated the norms of conventional society, it was as the heroic outsider or icono-clast. 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.[6]

According to Brand, Spacewar was not just a computer game but a kind of software samizdat, the vehicle through which the subversive hacker subculture was smuggled into the network of research laboratories sponsored by the Advanced Research Proj-ects Agency. The result was the creation of an increasingly global community of technician-radicals. Every night, "hundreds of computer technicians" in computer centers around the world engaged in an effectively out-of-body experience, "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 friends and wasting their employers' valuable computer time." These centers were anything but the isolated social wastelands por-trayed by Weizenbaum; rather, the computer center that Brand described constituted a vibrant social space, with its own "language and character, its own legends and hu-mor." 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 the Merry Prankster Acid Tests."[7]

These two radically different interpretations of the same phenomenon, as por-trayed by Weizenbaum and Brand, neatly capture the perplexed, ambivalent, and con-flicted attitudes toward 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 program-mers had no academic training and did not see themselves as scientists. (And, as the indignant and dismayed response of Weizenbaum to the computer bums clearly illus-trates, 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 analo-

---

[5] Stewart Brand, "SPACEWAR: Fanatic Life and Symbolic Death among the Computer Bums," *Rolling Stone*, 7 December 1972.

[6] Mitch Kapor and John Perry Barlow, "Across the Electronic Frontier," Electronic Frontier Foun-dation, 10 July 1990, https://w2.eff.org/Misc/Publications/John_Perry_Barlow/HTML/eff.html (ac-cessed 5 August 2014); Jonathan J. Rusch, "Cyberspace and the 'Devil's Hatband,'" *Seattle Univ. Law Rev.* 24 (2000): 577–98.

[7] Brand, "SPACEWAR," (cit. n. 5), 51.

gies and mixed metaphors: to many of its practitioners, programming was simultaneously art and science, high tech and black magic, work and play.[8]

If the discipline itself was opaque and incomprehensible to outsiders, so too were its practitioners. The colorful sobriquets invented to describe them—"wizards," "gurus," "computer boys," the "high priests of the new technology"—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 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."[9]

From a contemporary perspective, of course, the association of computer programming ability with a particular personality type is familiar to the point of being cliché. Today 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*.[10] 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).[11]

Indeed, in much of popular culture, the character of the computer nerd has 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 his contemporaries in the 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.[12] In the popular genre of steampunk fiction, the Industrial Revolution is reimagined as

[8] Nathan Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (Cambridge, Mass., 2010); Wendy Hui Kyong Chun, *Programmed Visions: Software and Memory* (Cambridge, Mass., 2011).
[9] Bruce Webster, "The Real Software Crisis," *Byte* 21 (1996): 218.
[10] Tracy Kidder, *The Soul of a New Machine* (New York, 1981); Steven Levy, *Hackers: Heroes of the Computer Revolution* (Garden City, N.Y., 1984); *WarGames*, directed by Wolfgang Petersen (Los Angeles, 1983), DVD.
[11] *Triumph of the Nerds*, directed by Paul Sen (New York, 2002), DVD; *The Social Network*, directed by David Fincher (Culver City, Calif., 2010), DVD.
[12] Neal Stephenson, *Quicksilver* (New York, 2003).

an abortive first attempt at the computer revolution, with Charles Babbage standing in for an early Alan Turing.[13] The perceived connection between 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 line of descent leading directly to the contemporary computer nerd.[14] The remarkable genius and accomplishments of Thomas Edison are now compared to those of Steve Jobs, and not the other way around.[15]

Like the 1970s-era computer bum, with whom he shares certain characteristics, the contemporary computer nerd is defined 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. The contested debate about the identity, expertise, and authority of computer programmers shaped many of the technical, managerial, and professional developments in electronic computing for the first several decades of the electronic computer era.[16] 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." This is not, of course, to suggest that women do not program computers; in fact, women played an unusually prominent role in the history of computer programming, especially in its earliest decades. And yet computer programming today is both male dominated and hypermasculine. 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 dismally low. The number of women studying computer science (as a percentage of total enrollments) has 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.[17]

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 that 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 antisocial, antisensual, and

[13] William Gibson and Bruce Sterling, *The Difference Engine* (New York, 1991).
[14] Benjamin Nugent, *American Nerd: The Story of My People* (New York, 2008).
[15] Walter Isaacson, *Steve Jobs* (New York, 2011).
[16] Ensmenger, *Computer Boys* (cit. n. 8).
[17] Thomas Misa, ed., *Gender Codes: Why Women Are Leaving Computing* (Hoboken, N.J., 2010); Jane Margolis and Allan Fisher, *Unlocking the Clubhouse: Women in Computing* (Cambridge, Mass., 2002).

attracted to the "hard mastery" of arcane technology pervades even the academic literature, most notably the influential work of Sherry Turkle, who provided the principal psychoanalytic framework through which the (male) nerd personality has been interpreted.[18] More recently, the perceived association between the "programmer personality" and mild forms of autism (to the point that Asperger's was sometimes referred to as the "geek disorder" or "Silicon Valley Syndrome") has reinforced the notion that there is a natural, historical, and inevitable connection between male forms of sociability and cognition and virtuoso computer programming ability.[19]

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 extraordinarily 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.[20] A remarkable demographic shift occurred in programming over the course of the 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 corporate hiring processes. In this respect, the history of computer programming provides novel insights into the structural factors through which the gendering of institutions and practices occurs.[21]

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 technological genre 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 re-created in other masculine spaces, such as bedrooms, basements, and dormitories. They persist today in the form of the corporate "campuses" (complete with "play areas," "nap rooms," and even "tree houses") of innumerable tech firms and start-ups.[22]

The final feature of this history concerns the ways in which male programmers mobilized multiple, and sometimes even competing, visions of masculine identity.

[18] Turkle, *Second Self* (cit. n. 3); Turkle, *Life on the Screen: Identity in the Age of the Internet* (New York, 1995).

[19] Jordynn Jack, *Autism and Gender: From Refrigerator Mothers to Computer Geeks* (Urbana, Ill., 2014).

[20] 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 (Chicago, 2001), 169–85.

[21] Margaret Rossiter, *Women Scientists in America: Before Affirmative Action, 1940–1972* (Baltimore, 1995); Ruth Oldenziel, *Making Technology Masculine: Men, Women and Modern Machines in America, 1870–1945* (Amsterdam, 1999).

[22] Katherine Losse, *The Boy Kings* (New York, 2012); Jim Edwards, "We're Jealous of This Start-up's Hammock-Filled Treehouse Office," *Fortune*, 14 December 2012; Ariel Schrag, "The Ping-Pong Theory of Tech Sexism," https://medium.com/matter/the-ping-pong-theory-of-tech-world-sexism -c2053c10c06c (accessed 8 August 2014).

Computer programmers might be predominantly male, but the masculinity of the computer nerd is hardly that of the police officer 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 a useful professional resource. 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.[23] Despite the 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 AND 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 Wajcman has called the "hidden history" of women in technology and seeking to engage in contemporary debates about declining female enrollments in computer science programs.[24] 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 a historical anomaly or a reflection of a temporary wartime shortage of men; rather, computer programming was a feminized occupation from its origins.[25] The use of low-wage, low-skilled female programming labor was integral to the design of early electronic computation systems. For the leaders of many of the pioneering computer projects, the assumption was that the process of "coding" a computer was largely rote and mechanical—and therefore work that could be best be assigned to women. Or, to borrow a relevant metaphor from computer programming itself, the presence of women in early computing was a feature, not a bug.

The realization that women were essential, not incidental, to the invention of computer programming turns on its head the conventional interpretation of its subsequent history, at least in terms of its gender dynamics. Programming was not born male, but rather had to be made masculine. It behooves us, therefore, to provide a brief outline of the feminized origins of the discipline.

The gendered nature of early computer work can be seen clearly in the US Army Electronic Numerical Integrator and Computer (ENIAC) project, arguably the most visible and influential of the pioneering wartime electronic computer development efforts. For the male leaders of the ENIAC, all of whom were scientists, engineers, or military officers (and, more often than not, all three), the important challenges as-

---

[23] Marie Hicks, "De-Brogramming the History of Computing," *IEEE Ann. Hist. Comput.* 35 (2013): 86–8.

[24] Judy Wajcman, "Reflections on Gender and Technology Studies: In What State Is the Art?" *Soc. Stud. Sci.* 30 (2000): 447–64.

[25] Nathan Ensmenger, "Making Programming Masculine," in Misa, *Gender Codes* (cit. n. 17), 115–41.

sociated with the development of a working electronic computer system involved the design and construction of the actual computer. The subsequent operation of the computer was considered to be relatively trivial, and therefore work that could be successfully delegated exclusively to 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.[26]

The sexual division of labor established at the ENIAC project provided a model for subsequent computer development projects. The very first written manual on computer programming, published in 1947 by Herman Goldstine and John von Neumann (and based on their experience with the ENIAC), 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.[27] The work of the "coder" was low-status, largely invisible, and therefore generally performed by women.[28]

Of course, the use of female labor to perform routine tasks was not peculiar to the ENIAC, or to electronic computing as a whole. The combination of mechanization, 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.[29] This was especially true of the "information factories" that emerged in the post–Civil War period. From the multidivision 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-skilled female labor.[30] In fact, by the beginning of the twentieth century, women dominated the clerical occupations. The reinvention of the electronic computer as a business machine in the postwar 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.

In any case, the association of computer programming with routine clerical work meant that, although computer programming in the 1950s was not a job performed exclusively (or even predominantly) by women, it was nevertheless gendered female. The assumed characteristics of programming work—routine, repetitive, and highly

[26] David Alan Grier, *When Computers Were Human* (Princeton, N.J., 2005).

[27] Herman Goldstine and John von Neumann, *Planning and Coding of Problems for an Electronic Computing Instrument* (Princeton, N.J., 1947).

[28] See, e.g., Jennifer Light, "When Computers Were Women," *Tech. & Cult.* 40 (1999): 455–83; and Nellwyn Thomas, "Selling the First Computer: The Legacy of the ENIAC's Publicity" (manuscript, University of Pennsylvania).

[29] Thomas Dublin, *Transforming Women's Work* (Ithaca, N.Y., 1994).

[30] Margery Davies, *Woman's Place Is at the Typewriter: Office Work and Office Workers, 1870–1930* (Philadelphia, 1982); Sharon Hartman Strom, B*eyond the Typewriter: Gender, Class, and the Origins of Modern American Office Work, 1900–1930* (Urbana, Ill., 1992).

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 than to men. So deeply entrenched was this association that in her book *Recoding Gender*, Janet Abbate quotes one female programmer who recalls being astonished even at the suggestion that the situation could be otherwise: "It never occurred to any of us that computer programming would eventually become something that was thought of as a men's field."[31]

In practice, however, the planner/coder distinction quickly broke down, and the work of the (female) coders became entangled with the intellectual operations originally carried out by the (male) planners.[32] This proved true even at the pioneering ENIAC project, where the tasks that the programmers performed turned out to be unexpectedly difficult, requiring the development of creative new techniques, further blurring boundaries between computer design and operation, hardware and software, and men's and women's work.[33] As more and more powerful computers were developed, the significance of software became even more apparent. By the early 1960s, companies like IBM and Remington Rand UNIVAC were manufacturing relatively low-cost electronic computers that were economically competitive with earlier forms of data-processing technology. But while the computers themselves could be mass-produced, the software systems that made them useful had to be custom developed.[34] Not only were there many more organizations in need of programmers, but 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 was largely used for scientific purposes, commercial machines were designed for business applications. The task of devising an algorithm capable of solving a differential equation, as challenging as that might be, paled in comparison to the complexity involved in constructing a computerized accounting system. 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.

This newfound 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.[35] 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 percent over the course of the next four to five years.[36] "Competition for programmers has driven salaries up so fast," warned *Fortune* in 1967, "that programming has become probably the country's highest paid technological occupation. . . . Even so, some companies can't find experienced programmers

---

[31] Paula Hawthorne, as quoted in Janet Abbate, *Recoding Gender* (Cambridge, Mass., 2012), 1.

[32] Jean Jennings Bartik, *Pioneer Programmer* (Kirksville, Miss., 2012); W. Barkley Fritz, "The Women of ENIAC," *IEEE Ann. Hist. Comput. 18* (1996):13–20.

[33] Thomas Haigh, Mark Priestley, and Crispin Rope, "Los Alamos Bets on ENIAC: Nuclear Monte Carlo Simulations, 1947–1948," *IEEE Ann. Hist. Comput.* 36 (2014): 42–63.

[34] Martin Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry* (Cambridge, Mass., 2003).

[35] Stanley Englebardt, "Wanted: 500,000 Men to Feed Computers," *Popular Sci.,* January 1965.

[36] Richard Canning, "The Persistent Personnel Problem," *EDP Analyzer* 5 (1967): 1–14.

at any price."[37] A talented programmer not only could command a high salary but also possessed an unusual degree of autonomy and mobility.[38]

The elevation of both the status and pay scale of computer programmers attracted a growing number of men to the occupation. Some of these men drifted in from disciplines with intellectual affinities to computing, such as mathematics, philosophy, or electrical engineering. Others entered via corporate computerization efforts and had backgrounds in traditional business specialties such as accounting.[39] In either case, these recent converts to computing brought with them the traditions, practices, and status hierarchies of their former disciplines, often attempting to re-create them in their newly discovered discipline. For these aspiring male programming professionals, 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.[40]

One strategy for dealing with this occupational insecurity was to emphasize the degree of skill required to be a successful programmer. If the problem with programming, at least from an occupational status perspective, was that it was considered to be straightforward and mechanical, then the solution was to reframe the occupation as being active, creative, and unpredictable. Given the growing scope of software projects in this period and the limitations of existing hardware, this reframing was not difficult to accomplish. Consider, for example, the work involved with writing a program for an IBM 650 computer (the first of the truly mass-produced computers, often referred to as the Model T of electronic computing). The main memory of the 650 was a rotating metal drum covered in magnetic oxides. Not only did the programmer have to analyze a complex business process and construct a program that automated its solution (using a limited and cryptic machine code instruction set), but, because magnetic drum memory was so slow relative to the 650's central processor, he also had to optimize the order and timing of critical operations to coincide with the exact moment that the desired data had rotated under the read head.[41] The difference between a program that ought, in theory, to work and one that actually functioned with an acceptable degree of performance was often a function of the singular skills and abilities of an individual programmer. No wonder they were so rare—and so valuable.

Getting a computer program to work properly under such conditions 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 to be proper engineering.[42] 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 innate genius (fig. 1). According to John Backus, the IBM researcher most famous for developing the FORTRAN programming language, programming in the 1950s was "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 primar-

[37] Gene Bylinsky, "Help Wanted," *Fortune* 75 (1967): 141–68.
[38] John Thompson, "Why Is Everyone Leaving?" *Data Management* (1969): 25–7.
[39] Thomas Haigh, "Masculinity and the Machine Man," in Misa, *Gender Codes* (cit. n. 17), 51–72.
[40] Edsger Dijkstra, "The Humble Programmer," *Comm. ACM* 15 (1972): 859–66; RAND Symposium, "Is It Overhaul or Trade-in Time: Part I," *Datamation* 5 (1959): 24–33.
[41] Mark Halpern, "Turning into Silicon: Further Episodes from Programming's Early Days," *IEEE Ann. Hist. Comput.* 14 (1992): 61–9.
[42] C. A. R. Hoare, "Programming: Sorcery or Science?" *IEEE Software* 1 (1984): 5–16.

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*, 13 May 1956, 157).*

ily on the programmer's private techniques and inventions."[43] While Backus did not intend this description to be complimentary—as an aspiring computer scientist he saw this reliance on individual ability and local knowledge to be demeaning—many other programmers saw this emphasis on personal creativity and esoteric skill as the source of their professional authority. To be a devotee of a dark art, 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 own domain.[44] It was certainly preferable to being characterized as a glorified clerical worker or a "mere" technician.

Anecdotal accounts of the unique genius of individual programmers were reinforced by an emerging empirical literature on programmer performance. In the late 1960s, the IBM Corporation commissioned a study (still widely cited today, despite its

[43] 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 (New York, 1980): 125–35.
[44] David Freedman, "Computer Magic," in *Proceedings of the 11th Annual Computer Personnel Research Conference* (New York, 1973), 1–9.

serious methodological shortcomings) that suggested that a truly talented programmer was at least twenty-six times more productive than his merely average colleague.[45] These exceptionally gifted "super-programmers" were "worth an army of programmers of lesser average calibre" argued one participant at a 1968 NATO conference on software engineering.[46] The conclusion drawn by many corporate managers was that "the major managerial task" they faced was finding and keeping "the right people": "With the right people, all problems vanish."[47] 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 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).[48] Women did continue to be hired as programmers in relatively high numbers, and through the beginning of the 1970s computer programming was regarded as unusually open to female participation, at least by the dismal contemporary standards set by comparable technical professions.[49] But there were also clearly masculine associations in the language and metaphors used to describe the distinctive 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.[50] When *Cosmopolitan* magazine published an article encouraging young women to pursue careers as "computer girls," the Computer Services Corporation, one of the largest employers of contract programmers in this period, published its own "humorous" defense of the inherent masculinity of their discipline.[51] In the advertisements from this period, women were often used as a visual proxy for low-skilled, low-wage labor. For example, if a computer manufacturer wanted to signal that its latest high-level programming language was easy to use, it would portray it being used by a female programmer—or, even more pointedly, a female secretary.[52] Such high-level languages were dismissed as "sissy stuff" by "real programmers" who preferred the "heroic" work of binary programming.[53]

---

[45] Hal Sackman, W. J. Erickson, and E. E. Grant, "Exploratory Experimental Studies Comparing Online and Offline Programming Performance," *Comm. ACM* 11 (1968): 3–11.

[46] Edward David, quoted in Peter Naur, Brian Randall, and John Buxton, eds., *Software Engineering: Proceedings of the NATO Conferences* (New York, 1976), 32.

[47] Robert Gordon, "Personnel Selection," in *Data Processing . . . Practically Speaking*, ed. Fred Gruenberger and Stanley Naftaly (Los Angeles, 1967), 79–88.

[48] Brendan Gill and Andy Logan, "Talk of the Town," *New Yorker*, 5 January 1957, 18–9; H. A. Rhee, *Office Automation in Social Perspective: The Progress and Social Implications of Electronic Data Processing* (Oxford, 1968).

[49] Lois Mandel, "The Computer Girls," *Cosmopolitan*, April 1967, 52–6; Richard Canning, "Issues in Programming Management," *EDP Analyzer* 12 (1974): 1–14; Bruce Gilchrist and Richard Weber, "Enumerating Full-Time Programmers," *Comm. ACM* 17 (1974): 592–3.

[50] Tine Kleif and Wendy Faulkner, "'I'm No Athlete but I Can Make This Thing Dance!' Men's Pleasures in Technology," *Sci. Tech. Hum. Val.* 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," *Tech. & Cult.* 38 (1997): 60–96.

[51] Mandel, "Computer Girls" (cit. n. 49); Computer Sciences Corporation, "In Case You Missed Our First Test . . . ," *Datamation* 13 (1967): 149.

[52] Ensmenger, "Making Programming Masculine" (cit. n. 25).

[53] Richard Hamming, *The Art of Doing Science and Engineering* (Australia, 2005).

Even the softer comparisons of computer programming to literary production—in his classic software engineering textbook, *The Mythical Man-Month*, Frederick Brooks famously compared programming to poetry—invoked traditionally masculine identities.[54] And the organizational role of "super-programmer," "hot shot," or "whiz kid" was likely more comfortable for men than for women.

By the end of the 1960s, a stereotype of the programming guru had emerged that was distinctively masculine. As the computer personnel consultant Richard Brandon described in a 1968 Association of Computing Machinery conference, the programmer type was "excessively independent," even 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."[55]

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 antisocial mavericks, at least by their white-collar coworkers: 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 unshaven and bedraggled (fig. 2).[56]

Possibly more significant was what Brandon described as the "Darwinian selection mechanisms" of computer industry hiring practices. By this he meant the industry reliance on psychometric testing, specifically aptitude tests and personality profiles, for the purposes of identifying trainees who possessed the "right stuff" to be skilled programmers. Such tests, which were used by more than two-thirds of all employers in this period, tended to filter for candidates who preferred to "work more with machines than with people."[57] After all, the widespread perception that computer programming was an innate and idiosyncratic ability, although conducive to the status and job security of individual programmers, provided little by way of practical guidance for an industry that suddenly found itself in need of hundreds of thousands of skilled professionals. 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.[58] 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

[54] Frederick Brooks, *The Mythical Man-Month and Other Essays on Software Engineering* (New York, 1982).

[55] Richard Brandon, "The Problem in Perspective," in *Proceedings of the 1968 23rd ACM National Conference* (New York, 1968), 332–4; Theodore Willoughby, "Are Programmers Paranoid?" in *Proceedings of the Tenth SIGCPR Conference on Computer Personnel Research* (New York, 1962), 47–54.

[56] These unusual hours often posed particular barriers to women, as many employers in this period had explicit rules against women being on the premises after hours. See Gerald Weinberg, *The Psychology of Computer Programming* (New York, 1971).

[57] W. J. McNamara. "The Selection of Computer Personnel: Past, Present, Future," in *Proceedings of the Fifth SIGCPR Conference on Computer Personnel Research* (New York, 1967), 52–6; Dallis Perry and William Cannon, "Vocational Interests of Female Computer Programmers," *J. Appl. Psychol.* 52 (1968), 31–5.

[58] G. Truman Hunter, "Manpower Requirements by Computer Manufacturers," in *Proceedings of the First Conference on Training Personnel for the Computing Machine Field*, ed. Arvid Jacobson (Detroit, 1955), 14–8.

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

Figure 2. Datamation *9 (1963): 42.*

fraction of the programmers required by industry. 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.[59] If the primary selection mechanism that they used to identify programming talent associated programming ability with a "detached" personality (read antisocial, mathematically inclined, and male), then it is no wonder that antisocial, mathematically inclined males became overrepresented in the programmer population, which in turn reinforced the original perception that programmers ought to be antisocial, mathematically inclined, and male.

Whatever the reasons for its origins, the association of masculine personality characteristics with innate and intuitive programming ability helped create an occupational culture in which female programmers were seen as exceptional or marginal. Like Edwin Boring's women Experimentalists, described elsewhere in this volume, only by behaving less "female" could they be perceived as being acceptable.[60] 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

[59] Datamation Report, "The Computer Personnel Research Group," *Datamation* 9 (1963): 38–9.
[60] See Alexandra Rutherford, "Maintaining Masculinity in Mid-Twentieth-Century American Psychology: Edwin Boring, Scientific Eminence, and the 'Woman Problem,'" in this volume.

confrontational interview techniques in which logic and math puzzles play a prominent role—despite the substantial evidence that such techniques are severely gender biased.[61]

IDENTITY CRISIS?

Of course, defining oneself in terms of esoteric genius or "rugged individualism" was not the only way to establish a professional identity. 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. In doing so, they mobilized a different set of masculine resources and rhetorics.[62]

As Margaret Rossiter and others have demonstrated, professionalization generally implies masculinization.[63] Consider, for example, the Data Processing 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 was particularly apparent: the requirement of formal educational credentials, a minimum of three years of 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.[64] The more computer professionals were seen as not only technical experts but also potential corporate managers, the more women were excluded. The man in the gray flannel suit might have occupied the opposite extreme from the bearded, sandal-wearing, programming guru, but they were sitting on the same spectrum of masculinity.

The principal 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 afterward 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

---

[61] 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* (New York, 2012).

[62] Nathan Ensmenger, "The 'Question of Professionalism' in the Computer Fields," *IEEE Ann. Hist. Comput.* 4 (2001): 56–73.

[63] Margaret Rossiter, *Women Scientists in America: Struggles and Strategies to 1940* (Baltimore, 1982); Jeffrey Hearn, "Notes on Patriarchy, Professionalization and the Semi-Professions," *Sociology* 16 (1982): 184–202.

[64] Marie Hicks, "Meritocracy and Feminization in Conflict: Computerization in the British Government," in Misa, *Gender Codes* (cit. n. 17), 95–114.

to 2:1, it imposed a strict four-year degree requirement for its members.[65] 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.”[66] There were even fewer female PhD mathematicians than there were women with undergraduate degrees.[67] 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 assuming an increasingly masculine identity. A survey from the late 1970s showed that fewer than 10 percent of ACM members were women.[68]

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 under way by the late 1960s and (*b*) it involved a turn toward the theoretical, the mathematical, and the abstract.[69] 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 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.[70] 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.[71]

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 uniquely 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 nascent masculinity 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 intuitive genius. What the academic computer scientist wanted was to establish his discipline on a firm foundation of theoretical knowledge.[72]

---

[65] Claudia Goldin, Lawrence Katz, and Ilyana Kuziemko, “The Homecoming of American College Women: The Reversal of the College Gender Gap,” *J. Econ. Perspect.* 20 (2006): 133–56.

[66] Editorial, “The Cost of Professionalism,” *Datamation* 9 (1963): 23.

[67] Margaret Anne Marie Murray, *Women Becoming Mathematicians* (Cambridge, Mass., 2001).

[68] Thomas D’Auria, “ACM Membership Profile Report,” *Comm. ACM* 20 (1977): 688–92.

[69] Michael Mahoney, “Computer Science: The Search for a Mathematical Theory,” in *Science in the Twentieth Century*, ed. John Krige and Dominique Pestre (Amsterdam, 1997), 617–34.

[70] Quote from Harold Sackman, “Conference on Personnel Research,” *Datamation* 14 (1968): 74–6, 81, on 76; Saul Gass, “ACM Class Structure” (letter to editor), *Comm. ACM* 2 (1959): 4; Anthony Oettinger, “On ACM’s Responsibility” (president’s letter to ACM membership), *Comm. ACM* 9 (1966): 545–6; “Why Are Business Users Turned Off by ACM?” (1974), CBI 23, “George Glaser Papers, 1960–1989,” Box 1, Folder 3, Archives of the Charles Babbage Institute, University of Minnesota, Minneapolis.

[71] D’Auria, “ACM Membership” (cit. n. 68).

[72] Dijkstra, “Humble Programmer” (cit. n. 40).

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 burgeoning 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 LABS AS SOCIAL SPACES

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, antisocial, 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 combination of technology and place, a combination that was specific 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, where the already gendered stereotypes associated with computer culture would become inextricably linked to adolescent masculinity, bright young students were allowed almost unlimited—and largely unsupervised—access to cutting-edge experimental electronic digital technologies. The norms, ethos, and practices established in the university computer centers of the 1970s formed the basis for the emergent computer hobbyist culture of the 1980s (and beyond) and would be perpetuated and re-created 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 spaces/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.[73]

   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.[74] 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 elec-

---

[73] Eric Raymond, "A Brief History of Hackerdom," in Raymond, *The Cathedral and the Bazaar* (Sebastopol, Calif., 2001), 19–64; Tim Jordan and Paul Taylor, "A Sociology of Hackers," *Sociol. Rev.* 46 (1998): 757–80.
[74] Grier, *When Computers Were Human* (cit. n. 26).

tronic computing was provided through these centers, rather than via traditional departments.[75]

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.[76] 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 the "service" connotations of the computer center and, indeed, from any association with actual computers.[77] 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 academic researchers had for the activities of the "computer bums" with whom these centers were increasingly identified.[78]

In its 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.[79] 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.[80]

[75] William Aspray, "Was Early Entry a Competitive Advantage? US Universities That Entered Computing in the 1940s," *IEEE Ann. Hist. Comput.* 22 (2000): 42–87.

[76] William Aspray and B. O. Williams, "Arming American Scientists: NSF and the Provision of Scientific Computing Facilities for Universities, 1950–1973." *IEEE Ann. Hist. Comput.* 16 (1994): 60–74.

[77] Atsushi Akera, *Calculating a Natural World: Scientists, Engineers, and Computers during the Rise of U.S. Cold War Research* (Cambridge, Mass., 2007).

[78] Michael Mahoney, "What Makes Computer Science a Science?" in *Science in the Context of Application*, ed. Martin Carrier and Alfred Nordmann (Dordrecht, 2010), 389–408.

[79] L. R. Fiock, "Seven Deadly Dangers in EDP," *Harvard Bus. Rev.* 40 (1962): 88–96; Anthony Chandor, *Choosing and Keeping Computer Staff* (London, 1976); Backus, "Programming in America" (cit. n. 43).

[80] Levy, *Hackers* (cit. n. 10).

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" peculiar to the university computer center—a culture so unusual that they thought it worthy of further study by anthropologists.[81] There was something "especially compelling" about the nature of computer programming, they argued, that absorbed its practitioners to such a 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 who threatened, rather than encouraged, 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 Zabarenko and Williams's 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 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*).[82] 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

[81] Lucy Zabarenko and Ellen Williams, "The Computer Center as a Subculture," *Council on Anthropology and Education Newsletter* 2 (1971): 5–8.
[82] Levy, *Hackers* (cit. n. 10).

"computer hackers."[83] 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 individual who found one-on-one interactions with a computer particularly compelling.

Second, 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.[84] This last was especially 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 codes that the computer bums obsessed over did not generally serve a pedagogical purpose and were rarely associated with their academic studies; in fact, the very best of the bums were notorious for not completing their course work, even when it related directly to their computer science curriculum.[85] 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 code, it was often to solve trivial puzzles or to tinker with programs that had already been written. The goal was not so much to accomplish an objective but to produce code that was beautiful, elegant, humorous, or otherwise aesthetically appealing. 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 line of code. Whether the resulting program ran quickly or efficiently, or even solved some useful or interesting problem, was irrelevant. The goal was simply to please oneself (as Stanford Professor John McCarthy described it, his students "got the same kind of primal thrill from 'maximizing code' as fanatic skiers got from swooshing frantically down a hill") or, more frequently, to impress one's peers.[86] A 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."[87] These listings would be passed around the computer center to be shared, admired, envied. Trimming code served as a form of masculine competition, a means of both demonstrating mastery over the machine and establishing dominance within the community hierarchy.[88]

[83] Ibid.
[84] Roy Rosenzweig, "Wizards, Bureaucrats, Warriors, and Hackers: Writing the History of the Internet," *Amer. Hist. Rev.* 103 (1998): 1530–52; Sam Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software* (Sebastopol, Calif., 2002).
[85] Richard Hamming, "One Man's View of Computer Science," *J. ACM* 16 (1969): 3–12.
[86] McCarthy, quoted in Levy, *Hackers* (cit. n. 10), 13.
[87] Levy, *Hackers* (cit. n. 10), 32.
[88] Ibid.

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 argued that the sociability of the computer lab was the key to effective learning and innovation in computer programming. In his study of the sociology of computer labs, he found that even small perturbations in the social and spatial networks of the center (e.g., the relocation of the soda machine) proved disruptive to learning and productivity. Programmers learned through conversation, by watching one another code, and by 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."[89] 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. Among other things, they could develop competitive games such as Pong and Spacewar and then play against one another.[90] 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 the abstract and disembodied activities of the computer hacker and more traditional, physical forms of masculine competition.[91] Finally, these graphical displays could be used to display pornography. The earliest documented computer "girlie pics" date from the mid-1950s, but no doubt these were 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 reproduced and/or cited in hundreds of academic papers.[92] Looking at "girlies" on computer screens (as opposed to, e.g., pursuing them in real life) might be a pathetically compensatory

[89] Brand, "SPACEWAR" (cit. n. 5).

[90] Henry Lowood, "Videogames in Computer Space: The Complex History of Pong," *IEEE Ann. Hist. Comput.* 31 (2009): 5–19.

[91] Gitte Jantzen and Jans F. Jensen, "Powerplay—Power, Violence and Gender in Video Games," *AI & Soc.* 7 (1993): 368–85; J. Jansz, "The Emotional Appeal of Violent Video Games for Adolescent Males," *Comm. Theory* 15 (2005): 219–41.

[92] Jaime Hutchinson, "Culture, Communication, and an Information Age Madonna," *IEEE Professional Comm. Soc. News.* 45 (2001): 1–7.

and adolescent masculinity, but it was masculinity nonetheless.[93] And at the very least, sharing such images with your friends in the computer lab created yet another opportunity for male sociability.

Compared to other places where young men would go to prove themselves—say, for example, the lofty peaks pursued by the Victorian mountaineers—the computer center might seem a vastly inferior alternative. And yet, as Michael S. Reidy describes in this volume, even in environments that were inherently dangerous, risk was socially constructed (by limiting one's food supply, avoiding the beaten paths, pushing the body's physiological limits).[94] In a similar manner, by engaging in marathon coding sessions, surviving for days on Cokes and junk food, and otherwise denying themselves, computer programmers could also engage in manly demonstration. In *Second Self*, Turkle describes a practice known as "sport death," in which computer programmers challenged one another to push the limits of sleep deprivation. As one of her MIT hackers described it, the "essence of sport death is to see how far you can push things, to see how much you can get away with. . . . I generally wait until I have to put in my maximum effort and then just totally burn-out."[95] And in fact, although Turkle does not seem to be aware of this, the concept of sport death was imported into the MIT computer center by a geology student, who had picked it up from rock climbers and parachutists at Yosemite.[96] The physical risks of computer programming might have been artificial and contrived, but they were nevertheless a form of masculine competition and display. In a rare moment of self-reflection about the gendered nature of such practices, the same hacker who described the phenomenon to Turkle noted, seemingly as an afterthought, that "women are not so into sport death."[97]

To the degree that the computer center was a social environment, however, it was almost exclusively a homosocial environment. Again, this is a stark contrast to the corporate computing experience. Although the stereotype of the bearded, besandaled computer programmer was well in place by the late 1960s, in actual practice women were still very much present in most corporate computer departments. There were certain rare circumstances in which women were explicitly excluded from the sanctum sanctorum of the corporate computer center (generally after hours, and then ostensibly to protect their personal safety), but in most corporations women represented at least 25–30 percent of all computer personnel.[98] If we include computer operators and keypunch operators (by then the most feminized of computer specialties), then the representation of women would be even higher. Not so in the university computer centers, particularly during the overnight hours—which is when most of the interesting action occurred. At these moments, the computer center was effectively males only.[99]

In part this was simply a reflection of the demographics of the student population—at

[93] Ben Edwards, "The Never-Before-Told Story of the World's First Computer Art (It's a Sexy Dame)," *The Atlantic*, 24 January 2013, http://www.theatlantic.com/technology/archive/2013/01/the-never-before-told-story-of-the-worlds-first-computer-art-its-a-sexy-dame/267439/ (accessed 5 August 2014).

[94] Michael S. Reidy, "Mountaineering, Masculinity, and the Male Body in Mid-Victorian England," in this volume.

[95] Turkle, *Second Self* (cit. n. 3), 194.

[96] Pepper White, *The Idea Factory: Learning to Think at MIT* (New York, 1991), 299.

[97] Sherry Turkle, "Computational Reticence: Why Women Fear the Intimate Machine," in *Technology and Women's Voices: Keeping in Touch*, ed. Cheris Kramarae (New York, 1988), 41–60, on 45.

[98] Sherry Turkle, "Advanced Programmers, Women Employment Seen Rising," *Datamation* 10 (1964): 69; Richard Canning, "Issues in Programming Management," *EDP Analyzer* 12 (1974): 1–14.

[99] Levy, *Hackers* (cit. n. 10), 75.

some of the 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.[100] As Douglas Thomas has suggested, university computer centers, and the hacker culture that emerged out of them, are examples of what Anthony Rotundo, in his history of American masculinity, has called "boy culture."[101] In such cultures, both affection and mastery is expressed through "friendly play," "rough hostility," and "affection through mayhem."[102] In the absence of opportunities for physical conflict, hackers turned to pranks, trash talk, and other forms of emotional aggression as a means of establishing masculine identity.

The Duke of Wellington famously ascribed his victory at Waterloo to the manly virtues acquired by his officers on the playing fields of Eton. Similarly, we might argue that the start-up culture of Silicon Valley was conceived in the computer labs of Stanford and MIT. These academic computing centers served as key sites of play and learning, central nodes in the informal networks of knowledge exchange that defined computing practice in this period, and obligatory passage points for the emerging hacker community. The historical exclusion of women from these environments, and the continuing gender specificity of their more modern equivalents, is therefore of profound and lasting significance. Reforming the culture means re-creating the spaces and places in which that culture is reified and transmitted.

TRIUMPH OF THE NERDS?

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 this 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 classroom-oriented "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 endured 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. And more often than not, these sessions were happening not in a university computer lab, but in the homes and bedrooms of the latest incarnation of Weizenbaum's compulsive programmer, the so-called computer hacker.

[100] Karen Frenkel, "Women and Computing," *Comm. ACM* 33 (1990): 34–46.
[101] Douglas Thomas, *Hacker Culture* (Minneapolis, 2002).
[102] Anthony Rotundo, *American Manhood: Transformations in Masculinity from the Revolution to the Modern Era* (New York, 1994). Quoted in Thomas, *Hacker Culture* (cit. n. 101), 19.

While the computer hacker bears some resemblance to the computer bum (indeed, Stewart Brand had deployed the two terms almost synonymously), the two figures are not identical. The computer bum was intimately associated with the university computer center (the only place where there were computing resources to bum); the computer hacker increasingly had access to his own machine, often in the privacy of his bedroom. Absent from hacker culture were the mediating influences of employers, faculty advisors, or professionally minded colleagues. Whereas the computer bum might have been a pathetic, wasted figure, the computer hacker was tinged by an element of danger. Prior to 1983, the word "hacker" appeared only infrequently in the literature; within a year, hacker was a household term. And in almost every case, the concept of the hacker was associated, if only indirectly, with the emerging problem of computer crime.[103] The most notable instance of this was the 1983 film *WarGames*, in which a young computer genius in pursuit of a video game brings the world to the brink of nuclear annihilation. But real-world examples of computer hacking, most notably the exploits of the "414 gang," who infiltrated computers at Los Alamos National Laboratory, the Sloan Kettering Cancer Center, and the Security Pacific Bank, suggested that "computer security" (as the problem would eventually come to be known) was a serious and growing threat to technological, economic, and even national stability. Hackers, unlike bums, were potentially malicious. Suddenly the notion of an obsessed, socially maladjusted young man armed with a powerful computing device didn't seem quite so harmless.

It is beyond the scope of this article to do more than sketch a brief outline of the computer hacker. Like the earlier stereotypes of the singularly gifted computer genius, the hacker is young, white, male, and focused on the computer to the exclusion of other interests. Youth (and maleness) had always been a defining feature of the popular conception of the computer expert (although the "boys" in "computer boys" was more often an expression of derision than a demographic reality), but the computer hacker, as constructed by sensationalist media of the 1980s, was almost by definition an adolescent male. It was a rare article on the growing incidence of hacker-driven computer crime that did not mention, generally in the first sentence, the age of the alleged perpetrator, and the younger the better.[104] Of the two central protagonists of journalist Steve Levy's *Hackers* (first published in 1984), one is a physically underdeveloped freshman at MIT, and the other is a twelve-year-old boy who happens to wander into the MIT computer lab. And while the statistics might have shown that a substantial number of computer programmers were neither young nor male, movies like *WarGames* provided a visual guide to what a "real" computer hacker looked like.[105] Even the dominant psychological explanation of the hacker mindset, which was Freudian, explicitly excluded women![106]

The adolescent male hacker introduced yet another layer of masculine identities

---

[103] Nathan Ensmenger, "From Whiz Kids to Cybercriminals: Emerging Narratives of Risk in Computer Security," *IEEE Ann. Hist. Comput.* (forthcoming).

[104] See, e.g., Paul Ciotti, "The Hacker's World," *Los Angeles Times*, 14 November 1988, which opens with the subheading "Young and gifted, they aren't afraid of getting emotionally involved with their computers," or Michael Schrage, "Teen-Computer Break-Ins: High-Tech Rite of Passage," *Washington Post*, 21 August 1983, among many others.

[105] As the movie poster for *WarGames* (cit. n. 10) makes clear, it is a young man who is the master of the machine. The only female character of any significance in the film stands behind him, looking over his shoulder in awe and admiration.

[106] Turkle, "Computational Reticence" (cit. n. 97).

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 hacker culture were actually well defined a half-century earlier by ham radio operators, for similar reasons and via similar processes.[107] Douglas in particular ties these practices to the late nineteenth- and twentieth-century "crisis of masculinity," in which young men, struggling to define themselves in a white-collar information economy in which physical strength and courage were largely irrelevant, turned to the mastery of technology as a means of demonstrating their fitness and potential. By engaging in ritualized forms of competition—in the case of early amateur radio operators this meant playing pranks on commercial and military operators, for personal computer enthusiasts "hacking" into computer systems—these young men could participate in controlled (and often socially approved, or at least condoned) acts of juvenile rebellion. The same skills and abilities that lent an edge of danger to the computer hacker or the phone "phreaker" were also those that could land him a high-paying job—or, by the late 1980s, turn him into a personal computer industry millionaire. Many parents were willing to risk the vaguely defined legal consequences associated with adolescent hacking in exchange for the opportunity for their sons to become the next Bill Gates or Steve Jobs.

What is most significant about hacker culture is that it was hegemonic. Although most computer programmers, then and now, did not consider themselves to be hackers, it became increasingly difficult for them to distance themselves from the connotations associated with popular representations of hacker culture. This appears to have been a particular problem for female programmers. Figure 3 shows a time series representing the percentage of women enrolled in undergraduate computer programs. As you will note, the graph demonstrates the notorious "bump" that occurs in the early 1980s: prior to 1983–4, female enrollments in computer science had been gradually increasing; in the years following, enrollments have, on the whole, continuously declined. There are many explanations for this decline, including a failure of the STEM education pipeline, a lack of female role models, and institutionalized sexism in higher education.[108] When we overlay the figures on female computer science enrollments with a representation of the rising number of media mentions of the word "hacker" (as measured by Google Ngram) we can visualize an alternative explanation.[109] The two graphs form an almost perfect inverse of one another. As the hacker stereotype came to dominate the popular image of what computer programmers do and who they are, they marked computing as an almost exclusively male domain. As Tove Hapnes and Knut Sorenson have argued in their study of Norwegian hacker culture, for many women in computing, the concept of 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

[107] Susan Douglas, *Inventing American Broadcasting, 1899–1922* (Baltimore, 1987); Kristen Haring, *Ham Radio's Technical Culture* (Cambridge, Mass., 2006).
[108] Caroline Clark Hayes, "Computer Science: The Incredible Shrinking Woman," in Misa, *Gender Codes* (cit. n. 17), 265–74.
[109] Google Ngram Viewer, https://books.google.com/ngrams/graph?content=hacker&year_start =1966&year_end=2000&corpus=15&smoothing=5&share=&direct_url=t1%3B%2Chacker%3B %2Cc0 (accessed 4 August 2014)
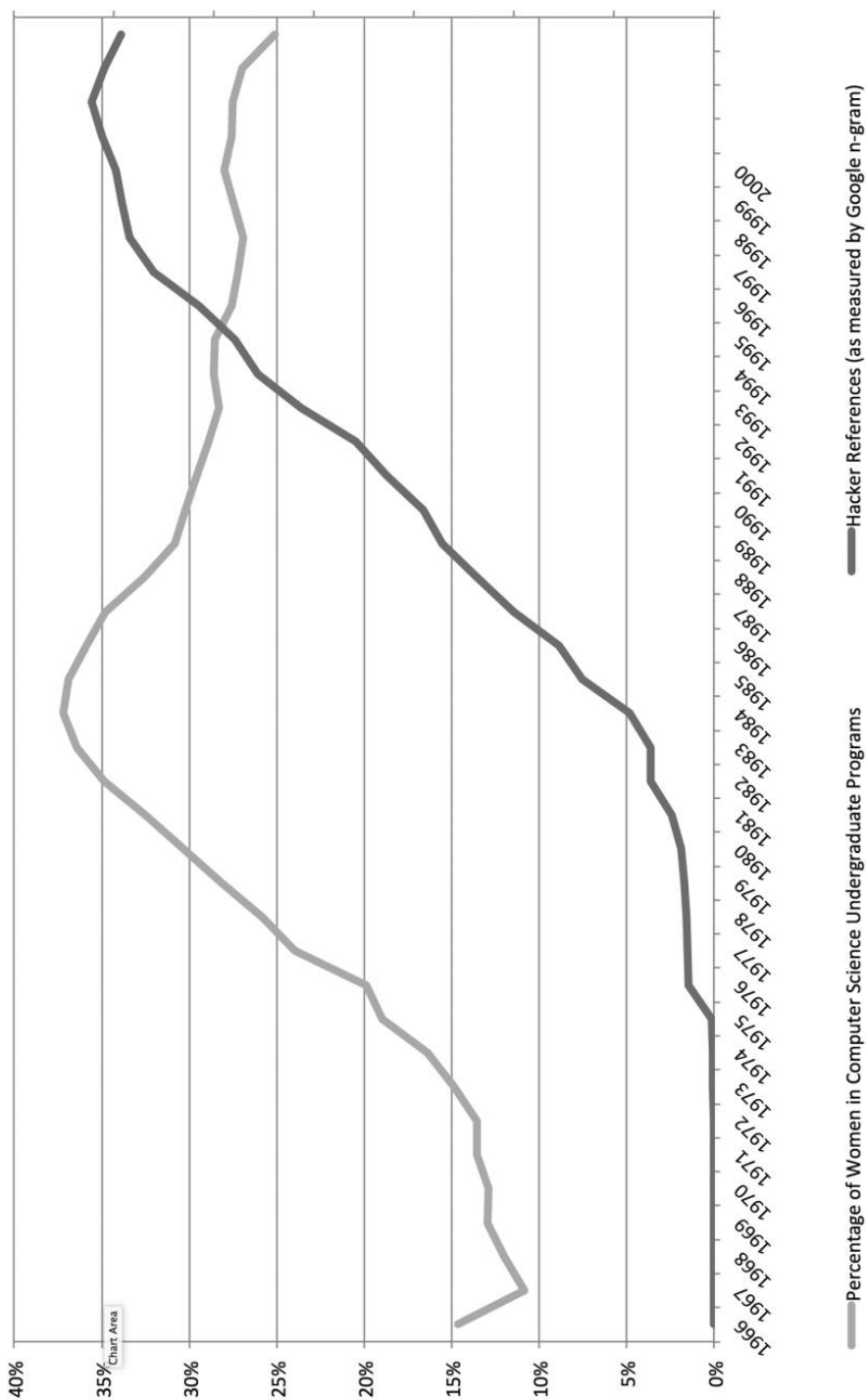
Figure 3. *Female enrollments in undergraduate computer science climbed steadily until 1982, when they suddenly started declining. For years the explanation for this has been something of a mystery. The dramatic rise in media representations of the computer hacker might be the explanation. The data on female enrollments are from the National Science Foundation. The references to the word "hacker" come from the Google Ngram viewer (National Science Foundation, Science and Engineering Degrees: 1966–2006 [Arlington, Va., 2008]; Google Ngram viewer, https://books.google.com/ngrams/ [accessed 4 August 2014]).*

problems with no obvious relation to the outside world."[110] The same origin myths and "triumph of the nerd" fairy tales that male hackers find comforting and empowering are, for many of their female counterparts, profoundly limiting narratives.[111]

It is important to note that many male programmers are also uncomfortable with the hacker stereotype. Not only is it not an identity available to every man (as Ron Eglash has described, it is particularly problematic for African-Americans), but it also precludes other forms of masculine and professional persona.[112] For those who aspired to be computer scientists or software engineers, the character and habits of the hacker were an embarrassment. But such is the nature of hegemony; by the middle of the 1980s, even for those programmers who aspired to more conventional professional identities, the emerging stereotype of the undisciplined computer bum provided a necessary 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 construction of one's chosen disciplinary agenda.

CONCLUSIONS

In Turkle's now-classic analysis of computer culture, she provided a psychoanalytical interpretation of the compulsive computer user. The obsessive computer user, according to Turkle, was a kind of paradox, a "loser" who saw himself as elite. Overwhelmed by the complexity of real-world social interactions, he retreated into the controlled, predictable microworld of the electronic computer. In the chapter that deals most directly with hacker culture and mentality, Turkle focuses specifically on the failure of the hacker to come to terms with his own masculinity. The chapter opens with a description of the annual "ugliest man on campus" competition, in which male MIT students flaunted "their pimples, their pasty complexions, their knobby knees, their thin, underdeveloped bodies." Turkle focuses on the contrast between the self-perception of the MIT students ("Everyone knows that engineers are ugly. . . . [That] to be at MIT is to be a tool, a nerd, a person without a body") and what they imagine to be a more ideal form of masculinity ("To be at Harvard is to be a gentleman, to be sexy, to be desired"). In Turkle's influential interpretation, the computer hacker is defined in large part by his lack of masculine identity. Hackers are good with machines for the same reasons that they are unsuccessful with people (and, in particular, women). However, in the context of their interaction with "the intimate machine," their tendency to be antisocial, antisensual, and overly focused on control may be transformed from liabilities into assets.[113]

To argue that computer hackers have constructed for themselves a "world without women" is not to suggest that they are not deeply invested in their own masculine

[110] 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*, ed. Keith Grint and Rosalind Gill (London, 1995), 174–91.
[111] Robert Cringely, *Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date* (Reading, Mass., 1992); Alison Adam, "Hacking into Hacking: Gender and the Hacker Phenomenon," *ACM SIGCAS Computers and Society* 33 (2003): 3.
[112] Ron Eglash, "Race, Sex, and Nerds: From Black Geeks to Asian American Hipsters," *Social Text* 2 (2002): 49–64.
[113] Turkle, "Computational Reticence" (cit. n. 97).

identity, however.[114] As I have attempted to demonstrate, the practices of "bumming," pranking, and other forms of technical display that originated in the university computer labs of the 1970s form the basis for a rich culture of masculinity within computing communities. Some of the most conspicuous features of this masculinity, and in particular the association of computer programming with the "computer nerd" personality type, are not so much a reflection of the essentially gendered nature of the activity (or, as Turkle suggests, the uniquely "intimate" nature of the technology) but are instead the by-product of attempts by early programmers to elevate the status of their discipline. In a wide variety of periods and contexts, from the corporation to the academy to the computer center, male programmers have mobilized masculinity as a means of pursuing professional status and autonomy. Many male programmers saw the role of the eccentric and exceptional 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 boy," most particularly the stigma of being narrowly focused, antisocial, 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.[115] In fact, one might argue, contra Turkle, that computer programmers, rather than being insufficiently masculine, have elevated the performance of masculinity to an extreme.

[114] David Noble, *A World without Women: The Christian Clerical Culture of Western Science* (New York, 1992). See also Erika Lorraine Milam, "Men in Groups: Anthropology and Aggression, 1965–75" (in this volume), on the self-conscious performance of ritualized competition among academic and philanthropic "men in groups" during the 1960s and 1970s.

[115] Magali Sarfatti Larson, *The Rise of Professionalism: A Sociological Analysis* (Berkeley and Los Angeles, 1977).