Coding Social Reality: Space, Knowledge Production, and Gender in a Collegiate Computer Science Lab

Ellen Marie Jordan
Anthropology Department
ANT 490 (for Departmental Honors)
Graduating May 12, 2018
Submitted May 5, 2018
# Table of Contents

Introduction .............................................................................................................. 2

Methodology .......................................................................................................... 6

Chapter 1: Space .................................................................................................... 8

The Lab as Territory .............................................................................................. 12

Digital Space ......................................................................................................... 16

Mobility .................................................................................................................. 20

Chapter 2: Epistemology and Authority ............................................................ 26

“It’s Kind of Like Legos” ...................................................................................... 26

Screens On, Screened In ..................................................................................... 31

“You’re Hardcoding?” .......................................................................................... 39

“Welcome, Boys!” ............................................................................................... 49

Chapter 3: Gender .................................................................................................. 55

The ENIAC Girls – Today and Now ................................................................. 56

The Master Theorem ............................................................................................. 62

Discourse ............................................................................................................. 70

Conclusion .......................................................................................................... 81

Bibliography ........................................................................................................ 84
Introduction

Sadie, a sophomore at Franklin & Marshall College, recently posted an Instagram picture of herself at the Sophomore Declaration Dinner (hosted annually to celebrate the sophomore class’s collective triumph in deciding majors). In the picture, she is smiling prettily and holding a sign that declares her double major in Computer Science and French. A witty caption, written in computer code, accompanies the picture, as does a second picture — swipe to the left, and a snapshot of Sadie and four other Computer Science majors replaces the one of Sadie smiling. Sadie is crouched front and center in a stylish pose, surrounded by the others. She is the only woman in the picture.

During the lifetimes of most of the students at Franklin & Marshall College, Steve Jobs, Mark Zuckerberg, and Bill Gates have become household names, each linked to their respective products: Apple, Facebook, Microsoft. Each of these products have had far-reaching impacts on the lives of those in the West – for example, this project was written in Microsoft Word. Perhaps due to its ubiquity and utility, Computer Science has become a prestigious field in which to work. Computer Science (CS) majors can expect a salary of about $70,000 from an entry level job, and, with time, a mean annual salary of over $110,000. The top ten percent of the computer science workforce earn the most when compared to all fields of engineering, almost $175,000 (Michigan Tech).

As demonstrated in the opening anecdote, most of these salaried workers are male. Since the mid-1980’s, the number of women in the industry has steadily declined, even as the industry itself is still growing. In the last fifteen years alone, the percentage of women in the field has dropped from 35% to 25% (ComputerScience.org, 2018). This seems to be established before the students enter college – when surveying AP exams taken by high school students, this similar 4:1
gender ratio applies, which suggests female students might be influenced by factors such as the stereotypes of the field (i.e. “geeky”) or by experiences at home (Misa 2010: 26). A survey of home computer use has revealed that men are more likely to use computers than women, perhaps socializing children to understand the computer as a tool for men (ComputerScience.org). Anecdotal evidence supports this assertion – in her ethnographic work with “geek girlfriends” and their use of free (non-propertied) software, Karanović remarked that, “a girl watched her mother type and, afterwards, started pressing the keyboard keys herself” (2012: 197). Through this sly insertion into her conclusion, Karanović seemed to hint that when women use computers around the house, their young daughters emulate those practices. This would support the suggestion that young boys who see their fathers using the computer would similarly gravitate toward it. This socialization during childhood seems to be reflected in the gendered proportions of the field (Margolis and Fisher 2002: 4).

Socialization processes do not stop in early development, although early forms of socialization do continually influence the experiences of men and women alike as they mature. A study performed by BRAID has released preliminary data that suggests that collegiate experiences also shape female involvement in the field, as does how classes are marketed to students. Linda Sax, who leads the research team involved in this study, attributes female disinterest in the field to factors such as decreased confidence in abilities (when compared to those of their male counterparts), even if the men and women are achieving the same grades. Women also might find Computer Science useless if they do not learn it in a setting that emphasizes the social good that can be accomplished through coding, which Sax connects to a perceived lack of collaboration within the field (Vu, 2017). Even when women enroll in CS classes, then, does not mean that they will continue taking classes within the Department.
This past year has brought especial contemporary salience to this work, both nationally and on campus. The #metoo movement (in which women use social media platforms to publicly call attention to the harassment they have faced from men) has initiated a national dialogue concerning gender equality and the place of women in society. Workforce concerns have been especially prevalent in this dialogue (Brzezinski, 2017), especially from women in male-dominated fields such computer science (Corbyn, 2018; Wong, 2018). Campus culture, while not explicitly referencing the #metoo movement, is reflecting this concern. The College Reporter, which publishes news from campus, has published an increasing number of protests made by female students and women’s advocacy groups on campus regarding safety measures that have (or, have not) been taken in response to third-party regulation of fraternity parties (Frazer, 2018).

Within this contemporary frame, I developed an acute interest in the experience of women, especially those at the undergraduate level (like myself). As journalist Nancy Niemi notes, women make up more than half of the population of degree-earning undergraduate students (2018). However, as the women of Franklin & Marshall have repeatedly communicated, something is not right – there seems to be a disregard for their concerns regarding their safety and well-being on campus (Frazer, 2018). Studying the entirety of campus was unfeasible for the scope of this project, initially chosen for my Anthropological Methods class, but it informed my decision to ultimately choose to study the Computer Science lab.

Throughout this thesis, I engage the concepts of space, epistemology, and gender. Within the Department, the manners by which space is constructed and maintained thus influences knowledge production and perceived authority within the lab. This, in turn, has important implications for how gender is both constructed and communicated within the lab. Drawing heavily from Foucaultian discipline (1977) and Bourdieuan habitus (1977), as well as materialist
theory concerning modes and effects of production, I argue that the Computer Science Department at Franklin & Marshall College implicitly excludes female participation through the reproduction of a space that privileges and prioritizes male contributions to the social knowledge of the lab. It is important to emphasize that these processes of exclusion are not conscious on the part of the men in the space. Instead, female exclusion has been rendered “normal” through a history of male-centered knowledge production within the field of technology that has not acknowledged female involvement. This, in turn, socializes all those who enter to expect a heavily masculine environment.

I first turn to space and the importance it holds for the students who use it. The students are disciplined to understand the space as having a certain utility (Foucault 1977), which then influences their production within the space. This ties space to production and introduces the importance of spatial materialism (Soja 1989; Lefebvre 1986), which asserts that space informs production and, thus, the social interactions that arise between producers. I extend this analysis when examining digital space and its role and place within the systems of production being learned by the students. Finally, I analyze the mobility of those who use the space, introducing the notion that the space is “semi-fixed” (Hall 1966: 109-10) and allows for a hierarchized representation of authority to be displayed and communicated within.

Production itself is a large consideration of this piece – students are always creating code, either for projects or to complete problem sets. I incorporate Marxist materialism to evaluate how students’ engagements in material practices underlies social relations which create the larger, gendered implications I come to discuss. As I mention above, space itself has been conceived of as having a material effect – it provides the location of the production, so it becomes equally important when discussing the social relations which are generated and
maintained from this production. This neatly fits space and epistemology together within my argument and lets me extend my analysis to evaluate how a certain kind of space can engender a certain type of resultant epistemologies (or ideologies). Production also influences competency, which is directly related to the authority one has within the space (de Certeau 1984). It is here that hierarchy is established among students, and it is here where gender becomes especially salient.

My final chapter examines how the masculine nature of the lab is made tangible and is reinforced through habitus, a system of practices and representations – an “externalization of internality” (Bourdieu 1977: 75). According to the Bourdieuan framework, early experiences condition people to act in certain ways. Thus, practice is borne from conditions, the structures of which produce a habitus that mirrors the perceptions and experiences that came before it (Bourdieu 1977: 78). I borrow heavily from this framework when evaluating how and what type of knowledge is being produced in the space. Gender, in this case, would be the ultimate social relation created from this habitus, and would explain the way in which women could be uniformly, implicitly excluded from the field of study.

Finally, feminist discourse theory (McElhinney 2005) provides an additional framework through which to visualize this exclusion. Gender becomes indexed through certain discursive modes, which create distinctions among the students using the space. Through this indexing, and through the male-dominated atmosphere of the space, masculine discursive modes can be selected for, placing female students at a disadvantage. Moreover, the hierarchization process relies heavily on this discursive method and informs how social relations are seen within the space – failure to take part in this discursive mode, then, is a failure to have a social place in the lab, and to be outside consideration of authority and competency. Women, with their separate
discursive mode, are thereby excluded from contributing to the social knowledge of the space, and thus feel unwelcome – or have to work much harder to be granted the same amount of respect and authority in the eyes of their peers as their male counterparts.

Methodology

For the months of October and November, 2017, I spent at least one to two days per week within the main Computer Science (CS) lab, Stager 002. This decision was based on recommendations from a student taking classes in the department who herself used the space. I only observed on Mondays, Tuesdays, and Wednesdays due to personal schedule conflicts and after learning the peak hours of the lab. Most homework assignments were due either Tuesday or Wednesday at midnight, so the beginning of the week saw more students using the lab space to complete those assignments. Additionally, a Computer Science 1 (CS1) tutor was present and available every Monday from 7-10 pm, and CS2 and CS3 tutors were similarly available Tuesdays during that time frame. No tutors held drop-in hours on Wednesday nights.

Nonetheless, Mondays, Tuesdays and Wednesdays were the busiest nights in the lab. As a student, my personal schedule conflicts and class schedule prevented me from being able to attend class or lab sessions held by the department. This influenced my decision to study the spatiality of the after-hours lab space, away from the structure of the professor-led class.

While in the lab, I participated in participant observation and informal interviews with tutors and students. I premeditatively arranged three of the interviews; the rest were done spontaneously and informally within the lab. I also conducted interviews with students in Blue Line if they could not talk while in the lab (due to instead needing to prioritize their schoolwork/assignments). These interviews were recorded via the recording application on my
personal, password-protected smartphone. These clips were then exported to my personal laptop, and relevant parts were transcribed using Audible onto a Word document.

By the completion of the semester, I had interviewed a total of seven students: 4 men, 3 women; 3 tutors, 4 non-tutors; 5 CS3+ students, 2 CS1/CS2 students. Two of these students were non-white. These numbers do not take into account the casual conversations I found myself participating in within the lab. I also took some pictures of the space on my password-protected smartphone. Lastly, I would occasionally (and informally) share my tentative findings with either students within the department or hired software developers who graduated from similar departments to periodically check for any errors I might have made in the interpretation of my findings. All the students’ names have been changed for the final version of this ethnography.
Spatiality

At night, the basement of Stager is a gloomy sort of place, a dark, quieter reflection of the building above. Stager Hall, central to what many see as the main part of Franklin & Marshall College’s campus, houses classrooms and, on the third floor, the American Studies and History departments. The Economics and Math departments are also housed here, which encourages a general-purpose feel to the building – the accumulation of flyers posted in the building throughout the year advertise a wide range of topics, experiences, and highlighted speakers. Additionally, it is not uncommon to enter a classroom for a social science class only to encounter mathematical equations covering the dusty blackboard. Look outside a window, and there stands Blue Line, an on-campus café/social hub for students, notable for its cozy study space upstairs, which treats one to a panoramic view of campus.

Blue Line, and the view of campus, falls out of one’s sight as one moves down the stairs from the Stager lobby. Two right hand turns later, and one is in the Computer Science department. On any given night, half of the overhead lights are off, casting shadows over the closed doors lining the hall – offices, one assumes, for professors who only physically occupy these spaces during the day. At the end of this first hall lays a small lab space, light spilling from it into the hallway. This is the “Quiet Lab;” when uninhabited, one can see “CS” in simple blue letters inscribed wearily on the façade of the closed door.

By following the brightest – and usually, the only – light source, and occasionally by listening for conversation, one happens upon the main lab space, Stager 002. If one enters by the main door to Stager, this is a route that seems slightly labyrinthine in its obscurity. If one had no knowledge of the room being in that basement on that side of the building, one would have no inclination to explore to that reach of the building. That first moment, one is carefully treading
through muted, dimly lit halls – the next, one is advancing up the ramp into the main lab space, blinking as eyes adjust to the comparative brightness of the room. To those with an inclination towards claustrophobia, this moment often comes as a relief. There is also a more direct route: a side door that bypasses the trek throughout the dim, quiet halls, avoiding any encounter of the Quiet Lab. However, this door is inconveniently placed on the opposite end of the building from the usual student’s approach from the campus residential area. The only doors that remain reliably unlocked after hours are those at the main entrance, which encourages many students to go through the front.

Stager 002 itself is a study in beige, the dark Formica wood desks handsomely pairing with the bland walls surrounding them and the dark carpeting beneath. These desks face the front of the room; the door is situated at the corner, to the right of a wide white board, a single projector facing the usually pristine white face. A lectern and chair at the whiteboard’s leftmost corner face the desks. The door stands perpendicular to a chalkboard, the dusty green surface almost fading into the wall behind it. Across the room from the chalkboard are two windows — the blinds drawn over the glass are the same shade as the walls from which those windows were meant to provide relief. The windows are usually covered, preventing one from seeing out – and preventing someone who is outside the lab from looking in. A single item decorates the space between these windows: a printed piece of paper, taped to the wall, which asks users not to shut down computers due to the multiple background programs tracking data for “research purposes.”

The focal point of the lab is the desk area. While the desks initially appear as a normal, if not newer, version of the classroom desk, a closer inspection yields some key differences. Each desk has a monitor attached to its base, and a keyboard that lies on a tray underneath the hard top. By drawing out the keyboard, a transformation occurs – the top of the desk lifts away and
down from the keyboard, revealing a desktop within the desk. Most of the time, this transformation from desk into desktop is the first thing students do upon arriving to the lab, almost like a reflex. This first interaction between student and the classroom desks offers a clue to the most basic purpose of the lab space, as a space for work meant to be done on computers. In fact, it is rare to see a student working in the space without a screen open in front of him/her. Perhaps for this reason, or perhaps due to its obscure, relatively unknown location within the specific department, the lab is only used by those taking classes in the CS Department. There is an additional lab that lays mirror image to Stager 002 specifically for the Film & Media Studies students, and students who wish to use Stager as a space for studying often find an empty classroom aboveground, where their classes most likely take place.

This enclosure of the lab from the rest of campus, along with its configuration, has direct implications for the social culture of the lab. Space is the canvas on which social relations are made possible – it is “a product literally filled with ideologies” (Lefebvre 1976: 31). Not only do ideologies occupy it, space also houses the modes of production which give rise to social relations, according to materialist theory, as described in Soja (1989). These modes can be concentrated in areas within space, so where production occurs has a direct bearing on the social relations which rise from this production – thus, space is directly relevant in considerations of production and social relations. The town was distinguished from the country through modes of production – similarly, the lab is distinguished from Stager and campus in its entirety. Adopting a materialist stance along with a Foucaultian presentation of discipline and its effect on students’ behavior within the lab, I make the argument that the space in which the Computer Science (CS) students work directly influences the production of knowledge and subsequent hierarchization of the students (the producers) that occurs in that same space.
The Lab as Territory

The strict use of the CS Lab by CS students suggests that a certain type of space is being created there by the students who use it. This created space is perceived in a way that either is exclusive or inclusive to the perceiver. The desktops and the obscure location work to doubly enforce this – because the CS students know how to use all the resources available to them in the space, and because the location of the lab is not public knowledge, it becomes “their space.” Familiarity with this space then becomes a marker of belonging for those who use it regularly. When a CS3 student said he did not know there was a water fountain on the opposite end of the basement, in the Film & Media Studies Department, his peers erupted into laughter. How could he not know? How long had he been using the lab, again? Did he not realize that is where the coldest water comes from? The water fountain marks the space as the property of those who use it – territory in its most commonsensical form.

The desktop also acts to focus and center the students’ orientations within the space. If students under observation were working on individual laptops, they usually also had the desktop open, even if nothing important was on the screen. Paper might be spread around the desk, a laptop sitting cockeyed off to the side, yet the desktop usually remains the center of user attention. The students always return to the center to plug in the variables that they have coded and noted down on that strewn paper. Multiple windows might be pulled up on the screen – the colorful input window versus the more solemn black and white output, as is seen in Python, a basic computing program that is taught in CS1. While students work on the desktops, they primarily face these computer windows and the empty white board in the periphery of these screens, not each other; even if multiple people are collaborating on an assignment, screens stand between them and any person with whom they are talking.
The lab is then a territory, a “fixed-feature” space that governs the activities of those who use it (Hall 1966: 103). Through having class and/or lab in either that space or one similar to it, students become used to how they are expected to participate in the space. This orients the students to the space, a process that Hall justifies as necessary because “to be disoriented within a space is to be psychotic” (1966: 105). In other words, situating oneself within the space requires an understanding of the space and its purpose, yet this orientation also allows the student to understand his/her role within the lab. The space situates the person to others – by using the desktops in the Lab, a student is automatically considered to belong in the space, which is tantamount to saying to be part of the Computer Science Department. Training students how to better use and understand these desktops, then, is a process in which the students become more familiarized with the spaces involved. This means that “fixed-feature” space is “the mold into which a great deal of behavior is cast” (Hall 1966: 106). It precludes and is inherent in the actions taken by the individuals in that space.

Learning how to negotiate this fixed space then becomes a form of internalized discipline (Foucault 1975: 137). Students in the space are trained through class and through socialization within the Lab to orient their bodies and attention in certain ways. An example of this can be seen in the methods students employ to attract the attention of the tutor. Younger students automatically raise their hands to attract the attention of the tutor (an act that they may have learned as being polite and “right” through their previous experiences from primary education onwards). Older students, on the other hand, personally call on the tutor, who they treat as a peer, not a superior. This suggests that the discipline of the Lab is different from the discipline of the classroom – and brings with it at least some of its own rules, to which new students must be socialized. The progressive change in the formality of interaction between groups is codified in
the bodily gestures that are being learned by the students who use the lab. This also suggests that a hierarchical mode of power exists in the lab, which “targets” students who use the space to mold their behavior within the space to adhere to what is acceptable or normal (Foucault 1975: 137). Those who have not learned that calling out to the tutor is acceptable within the space are obviously novice to the space and its rules. How one uses the space then has implications for systems of hierarchized authority within the space.

Space has a direct bearing on the learning of discipline, as Foucault suggests. First, the Lab is enclosed – there are boundaries that mark its belonging to the Computer Science Department. The “CS” hanging on the door, for example, marks everything within that room to be under the aegis of “Computer Science.” Closing the window blinds, too, suggests a purposeful obscuration of the space – people walking past cannot look in; the space is not open. This creates a “protected place of disciplinary monotony,” in which students learn to focus on their schoolwork for Computer Science (Foucault 1975: 141). The Computer Science students do not use the time in the lab to work on assignments from classes outside of the Department, displaying a disciplined behavior within the Lab – the space is meant to be used with the chief purpose of working through assignments for Computer Science.

This docility and conformity in the use of space is mutually defined by social relations within the lab. The body-based minutia of behavior inculcated into these students manifests and collects as a composition of the social relations that fill the space and gifts it its specific character (Massey 1994: 22). To briefly return to Hall’s model, such social relations must be generated by the physicality and utility of the space itself. The “behavior” discussed above is not just technical (pertaining to the use and construction of code) but also always social. Discipline, in this case, acts as both academic discipline – knowing how to orient one’s self to code most efficiently –
and social discipline. This means that not to use the space is not to engage socially with the department. In the same vein, not to exhibit the same discipline in the space is not to fully belong to the space.

This is reflected in comments made by senior students who have been using the lab for years. Billy remarked that the space helped to bring people together to work on projects, continuing by saying that it was “lame” that not as many people as before, when the lab space was incepted, came to the workspace. When pressed to clarify, he stated that there was a “stronger” culture among the majors when more people used the space. In effect, he branded those who instead choose to code in their own rooms as outcasts. Being “lame” is not a desirable label from one’s peers, as Billy states. In fact, being “lame” implies that one is not doing what everyone else is (or, should) be doing – using the lab space after hours. Not only is the lab space a space in which sociality within a major can be experienced, it is encouraged, by students and professors alike, that the space is used.

This same sentiment was shared by Rahul on a separate instance, when he said that he only became close with other majors when he was taking CS3. At that point, evidently the material covered in class became so challenging that it necessitated the use of the lab. The space was given a purpose by those who sought a place in which they could find help with the challenging work: to facilitate the learning and subsequent success of the students who used it. Even before CS3, though, specifically in CS2 and CS1, traces of this are seen. CS1 students work mostly independently, sitting next to each other yet not speaking much, except to the tutor. CS1 students tend not to be seen in the lab space on nights when the CS1 tutors are absent. This pattern of use additionally emphasizes the space as one that has a specific utility – it should be
used for work done for Computer Science classes. The emphasis on the proper utility of the space, too, trains students how to behave and operate within the lab space (Foucault 1975: 143).

The shared utility of the space is not always so formal, like a classroom might be. By CS2, students begin to aggregate into small clumps to work and occasionally to chat. There is a familiarity and ease with each other in the lab that carries into CS3. The older students are the most comfortable with each other, which is represented through the relatively large groups formed for collaboration and the joking, informal nature of these groups. However, this social behavior is inextricable from the basic utility which students have been disciplined to associate with the space. Dave confided that he worked better in the lab on Friday evenings, when nobody else was using the space, because it was a more comfortable than his room, which would be just as quiet. In this case, it seems that he has been disciplined to expect to work best on Computer Science when in the space that specifically encourages this work.

Digital Space

The utility of the lab space is focused on the desktop. Using Foucaultian ideas about spatiality, we see that what he calls cellularity and partitioning are as important as enclosure in the discipline of the student (Foucault 1977: 143). The desktop partitions students, gives them their own space in which to work. Two students do not work on the same desktop, just as one student does not work on two desktops – although one student may work on both a (public) desktop and a (private) laptop. Not working on the provided desktop, then, is abnormal or odd. In the moments when a program on a computer is acting “buggy” or “slow,” the student often remarks on the situation, as if justifying why [s]he is not working as hard or effectively as [s]he could. Even if work is not actively being done on the computer, interacting with a computer
seemed to provide a basic justification for one’s presence within the lab. I myself only felt truly comfortable in the space if I had my laptop open in front of me, even if I was not using it – I felt less conspicuous than I did when I simply sat at the desk taking notes in my notebook.

Moments when time is not being spent doing work become moments in which social interaction among the students becomes a requirement of the lab. For example, in the time between turning on the computer and fully logging in, the student is facing a blank screen. Greetings and small talk take place in this interstitial moment, before the computer fully “awakes,” tapering off when the programs load and the student becomes enveloped in an onscreen world of his or her creation. In this moment, verbal acknowledgement explains why the student is not yet making use of the space. The screen, as it warms up before granting the student entry into its digital interior, reminds the student that [s]he is still within a different space, one that is more physical and geographic. Utility is linked to spatiality – the use of the virtual space inside the screen (for production within that space) is the purpose of the desktop. Using that space additionally suggests that one has left the physical space of the classroom and has entered another. Social interactions could happen here, via social media or email – but it is no longer necessary to be social with those in physical proximity to oneself. This is not to say social interactions cannot still take place. In fact, while it can feel like one completely leaves this space, that is not entirely true to reality – it is impossible to completely enter the digital space. Even when completely submerged behind the screen, one’s body still occupies the physical space of the lab.

While I primarily studied the utilization of the physical space of the lab, this is not to imply that the digitized realm is less real to those who work in it. In fact, the field of digital anthropology draws heavily from materialist theories. First, the digital world can only be
accessed through material infrastructure such as computers and smartphones (Miller and Horst 2012: 25). The content that is displayed and “shared” is further able to shape the actions and ideologies of those who access it – thus, human-technology interaction is a material practice. Miller and Horst go further to assert that this material practice then becomes socialized into something like the Bourdieuan habitus, which will necessarily reproduce itself (2012: 29).

Digital space, then, this realm of zeroes and ones, is an influential factor that materially shapes the lives of those who operate and situate themselves in proximity to the space – yet how does this digital space compare to the physical space of the lab?

To answer this question, it is necessary to develop an understanding of spatial materialist theory, as posited by Soja (1989). In his consideration of the role of spatiality in (then-contemporary) Marxist theory, he asserts that, “social relations of production are both space-forming and space-contingent” (Soja 1989: 81). Through a return to the analysis of modes of production and the social relations engendered by these modes, he manages to incorporate the reality of space into materialist conditions that had previously been less accepting of the concept. This is more broadly an extension of work done by Lefebvre, who argues that to truly understand capitalism one must first understand that it is dependent on “socially mystified spatiality,” veiled in illusion and ideology much like exchange has been (Soja 1989: 50). Space is not “innocent,” or neutral – it is political, historical, filled with ideologies (as well as objects). Therefore, it must be a social product in the way in which exchange is a social product (Lefebvre 1976: 31).

Furthermore, since space is a social product tied to modes of production, it is directly related to the creation of hierarchized classes. Lefebvre concentrates on the possibility of exploitation implied by spatial separation, citing colonialism as an example of this exploitative process (1976: 36). Where objects are produced thus has as much of a social implication as do
the mechanisms by which they are produced— in fact, where production happens becomes enfolded into how it happens. Soja discusses this as “space-to-class homology,” which relates the division of space into dominant centers surrounded by “subordinate peripheries” (2012: 78). Production occurring in the center, then, is privileged—this is where the bourgeois class proliferates, so to speak. To be in the center of productive space is to control modes of production and thus to control ideology itself (Marx 1978: 164).

Digital space neatly fits into this materialist concern with space when considering the labor that goes into the design and creation of digital space (Miller and Horst 2012: 26). However, digital space is additionally complicated through its abstraction—its abstract, virtual nature seems to render the space as less “real,” when compared to physical, geographical space. Production within this space has thus been separated from the more usual social production of knowledge. In fact, it is purposefully relegated to a select set of individuals: coders, or “geeks” (Miller and Horst 2012: 25). As this distinction has been habituated, production within digital space thus has been normalized to make sense when it is a coder who is producing—as a result, one does not regularly code (produce within digital space) without being considered a “coder.” A concentration of coders in one space then creates a hotspot of production—a central space of production. A central space of production then correlates with a central space in which ideologies are produced that will come to shape social behavior and expectations within the lab.

Students’ involvement with digital space is then a defining characteristic of the Computer Science lab. The unique material conditions of social production within the space serve to separate it from other spaces on campus, such as the Film and Media Studies Department. While both Departments share the basement of Stager, the lab spaces are separate, not interchangeable or shared. Stager 002 is always the Computer Science lab and Computer Science students are the
only students who use the space. While there is no explicit divide between the CS Department and the Film and Media Studies Department within the basement, students are disciplined to understand and operate within the forms of habitus established by modes of production occurring within the specific space. Still, space is never passive – the production at hand is constantly reproduced within a “dialectised, conflictive space” (Soja 1989: 50). The dialectical contradictions inherent in the space are what drive and reproduce the habituated modes of production – thus, the existence of the lab space actively encourages the discipline that allows for a disparity of production to occur between the lab space and the non-CS neighboring space.

Space usage within the lab additionally establishes an internal hierarchy that reflects production within the Department while more broadly distinguishing those who know how to use the space from the rest of the student body.

Mobility

Students are not just disciplined to understand the fixity and territoriality of the space of the lab – they also become disciplined in how to arrange and move within that space. The interactions described above imply a notion that challenges the fixity of the space – the space can be rearranged, or modified, to best suit an individual’s needs. These modifications then prompt interactions in marked ways (Hall 1966: 109-10). The best way to visualize this is in how a student establishes his/her study space within the lab. Papers might be set on the desk next to him/her, available in case the student needs to plan code or reference notes. By creating a larger radius of stuff around the desktop, the student is impeding others from sitting at those other desks, thereby minimizing the chance of socializing. If [s]he wishes to socialize or acknowledge
a friend entering the space, [s]he will decrease this radius by moving some of his/her things out of the way of the friend, inviting the friend to sit close.

Proximity thus becomes a precursor gesture signaling a willingness to socialize, although it may also require an invitation or some previous form of contact as well. Unless there are no other desks available, strangers do not generally sit next to one another. Proxemics, then, is at play here – the constructed distance between students communicates to those students how they should interact and communicates to other observers how those students feel about each other (Hall 1966: 114). If the students are not personally close, this is mirrored in their seating and work arrangements within the space.

The mobile chairs also allow for semi-fixedness within the space – a student can swivel to either side or move closer to another to collaborate with him/her. This is taken advantage of when working with peers in small groups. Students can also arrange themselves in clusters if they wish to work together, instead of sitting along a straight row – this allows for socialization and collaboration to occur with more people. More importantly, this allows for students to choose with whom they wish to socialize, or if they wish to engage with others at all. The room can then be altered in specific ways to best engage others – this process is learned and implicitly involves the same sort of proxemics that are important when students choose where to sit in the lab. This being said, the chairs remain in the lab – at the end of the night, the students do not wheel out into the hallway and back to their dorms. Instead, they stand and leave the chair in the lab. Standing, then, is not like sitting.

Mobility, naturally, differs according to the night. Tutor drop-in hours are strategically spread so that there is at least one tutor in the lab three nights a week. The tutor holding the drop-in hours seems to influence the make-up of the people who use the lab that night. For example,
on Monday nights, one of the CS1 tutors is present. Most of the people who then use the space at this time are in CS1 and are there to ask questions. On Mondays, too, one can see the most women using the space and the least number of men. Monday nights are quiet, lacking the more mobile energy seen and felt on Tuesdays and Wednesdays (when the CS2/3 tutors and no tutors are present, respectively). Mobility is then another discipline that needs to be learned in the lab, much like calling out to attract the attention of the tutor.

Those with the most mobility in the Computer Lab are the tutors themselves. When students have questions, they call the tutors to their workspace, so throughout the tutor’s drop-in hours he or she is kept busy walking around between students. Despite this mobility, the tutor bases him or herself either at one of the desks (signifying that he or she is just as much a student as everyone else), or at the podium at the front of the room, emulating the stance of a professor or supervisor. When [s]he is helping students, [s]he will sit on a desk neighboring the student, pull up a chair, or just stand behind the student. The student, though, rarely gets up, raising a hand or calling out for help. This fixity, the inability to fully mobilize throughout the space, begins to hint at a larger hierarchy being established. Older students will occasionally walk into the space to greet those who may be using it without using the space themselves, freely exercising their learned ability to be mobile. (And thereby expressing their hierarchical place through their mobility.) Billy, one night when Stager 002 was empty, walked next door to see if anyone he knew was working in the Quiet Lab. (He soon returned, shaking his head to convey that he was not successful in his endeavor). Through his comfort with the lab and the subsequent mobility that offered, he felt justified in being able to choose in which space he wished to be – and this choice was heavily contingent on the others in the space. Mobility is then part of a pattern of choice, in which students can select with whom they wish to socialize.
The mobility of the tutors is not quite so carefree. Instead, their mobility is meted out depending on the needs of the students. Theirs is a more supervisory role. The tutors ensure that all who come for their help are completing the problem sets in the correct manner. This inculcates in the students the discipline necessary to correctly produce within the digital space while also reproducing the notion that there is a “correct” method by which the problem sets need to be finished. Nowhere was this clearer than in an interaction between Charles, the CS2 tutor, and Nate, one of the students in CS2. When Nate asked if the code he had created would suffice – it did, after all, yield the correct effect – Charles advised him to substitute some of his functions for another type. According to Charles, the problem set was meant to test the ability to use the missing function – failing to include it would not be the correct way to complete the assignment. In this instance, Charles’s supervision helped guide Nate to correctly complete the assignment – thus, his supervision enabled the correct type of discipline to be learned and executed (Foucault 1975: 150). It also reinforced the disciplined practice of completing the Professor’s assignments to meet the Professor’s wishes, not the student’s. Here, Nate is reminded to keep in line with the expectations of the course. The lab space provides a place for this supervision. The lab space is then necessary for the particular processes of knowledge production which take place within the Department, which rise from the discipline of the student and his/her diligence with the assigned exercises.

This supervisory form of mobility is not practiced by all upperclassmen, which suggests that this inculcation of discipline is meant to come from a very specific place within the lab space. While disciplined exercise is reinforced through interactions among all the students using the lab, supervision of the exercises is left to those who move through the lab in that supervisory manner. Just like how most of the students using the lab do not sit at the front of the room facing
back towards the rows of desktops, some of the most senior students do not appreciate moving from their work space. Dave, when asked to come over to answer a question, warned that his friend better be asking a “good question” because that was the only way he could justify getting out of his chair. In other words, Dave’s purpose in the space is not to supervise the minutia of his friends’ problem sets, but to instead to help when there is a big, legitimate obstacle his friend needs help overcoming. The tweaking of the exercise is not his intent – his place in the space is not to teach discipline, but to reinforce it. By this, I mean that his expectations and participation in the space reinforce what is “normal” in the space without imposing any new exercises or discipline on his peers in the space.

Mobility within the space then implies a hierarchy, which, as we shall see in the next chapter, is contingent on how an individual’s knowledge production is perceived within the lab. Those who perceive themselves to have utmost authority have no qualms walking throughout the entirety of the lab space, looking over the shoulders of other students and making remarks that indirectly hint at the students’ discipline. This may harken back to a form of habituated mobility seen in the classroom – the pacing and supervision of a Professor, while missing from the after-hours lab, is suggested in the prowling of tutors and in the tentative hand raising of young students.

On quiet, cool nights, one can hear the hum of the two radiators begin and end at regular intervals. (These, while providing a background to the conversation, also manage to heat the lab to the point where sweaters in December are no longer necessary.) Sometimes, an odd gurgle from a coffee machine somewhere outside the lab intrudes into the space. It is not always a quiet space – people are free to talk as they wish (or not), although the modes and styles of discourse
in the lab are influenced by social relations that arise in the lab. These social relations are enabled through a process of disciplining students to the behaviors within the space, fostered through the enclosure of the space away from the rest of campus and the cellularity of the desktop. Furthermore, these social relations are the direct product of modes of production fostered through use of both physical and digital spaces. This would not be possible without a space to call their own, albeit one nested underground, out of sight from campus yet lying at its heart.
Epistemology and Authority

“It’s Kind of Like Legos”

The greatest challenge of a tutor is to facilitate understanding on the part of the student. What analogy would be most conducive to help the struggling student? How many times does a certain topic need to be revisited before the student seems to truly understand? As a tutor in the Biology Department, these questions are familiar ones to me. While interviewing and observing tutors within the Computer Science Lab, I did notice that the tutors alter approaches along similar lines. Sometimes, a tutor would stand up and begin writing equations and functions on the board – other times, he or she would be content to provide a verbal explanation of the material to the student.

This variability in approach suggests at least a subconscious understanding on the part of the tutors about the differences in student learning styles, especially when it came time to apply that learning to problem sets asking students to accomplish a set number of goals. Yet when I held interviews with these same tutors, many expressed to me that their goal is not so much to help the student understand the piece of syntax that might be confusing them than it is to help them with basic problem solving. Rahul, for example, identified the truly challenging part of tutoring to be the responsibility he feels to teach the students how to think, not just what to do when they encounter a certain problem. Coding, he alleged, is abstract and only requires a capability to problem solve. The syntax is the “easy stuff” – a quick Google search can yield a function or phrase within milliseconds. (That being said, how long will it take before coding becomes such an encoded process that AI could take over, rendering coding knowledge obsolete?) Billy affirmed this separately. As a TA for CS1, he said he witnessed a “divide” occurring within the class between people who “got” that way of thinking and those who struggled with translating the concepts into coded product.
While this comment seems to reaffirm the notion that how one naturally thinks is what determines one’s success in the Department, it does gloss over the reality of coding. The practice of coding is not easy – talk to any CS student for long enough, and [s]he will acknowledge that. Basic problem-solving skills, ones that non-coders are equally likely to possess, are not the only prerequisites needed to excel in the field. Students have to be taught how to problem-solve in the context of the lab – and here, knowledge production begins in earnest. Problem solving via coding requires the ability to work toward an expected goal – a finished code which is capable of enacting some expected effect – using given “languages” and functions. This requires the ability to think within given parameters to create, a process that Billy calls “mathematical creativity.” This often requires the students learn to situate themselves within the given parameters of the coding “rules” to achieve that which they wish.

Allie had been talking to the two CS1 students for about ten minutes before she stood up and walked to the board. She quickly scribbled a function onto the board, the chalk making muted squeaking noises as she wrote. Then she turned around, chalk in hand, and said, “So, now you’re in the ‘while’ loop. What now?” By her prompt, the students began to brainstorm, throwing out tentative suggestions for where she could “go” next.

By situating the students within the coding process, Allie was prompting them to problem solve their way out of a coding dilemma and to push beyond it. If coding in general is to be understood as an extension of problem solving, then that problem solving seems to be contingent on the ability for students to situate themselves within the screen, making themselves more present and active within the digital space in which they must work – this process is explored more fully in the next section of the chapter.
It is this process, though, that develops the proper sensitivities and skills of the coder. As students grapple with problem sets in their classes, they are exposed to levels of increasing complexity in coding. A CS1 final project might be to design a quiz that predicts which college major would be a good fit for the quiz-taker. A senior capstone project might be the construction of a code that works like a human neural network and can predict emotion when given a “tweet” from which to work. The latter is considered (by professors, other students, and society outside of the lab) to be of more worth. Although the student’s time investment might not have been that disparate between the two, the amount of expertise needed to create the latter would be greater by several magnitudes. This will also be further explored in this chapter.

Understanding the manual process of learning to code is critical for understanding the workings of the lab. The lab is, after all, a space in which students can practice, maintain, and learn skills needed to excel in their chosen field of study. The technical skills then allow for forms of hierarchy (which are strongly gendered) to be reproduced, so the lab is additionally a space in which knowledge production is intimately tied with the practice, maintenance, and learning of gendered dynamics within the field itself. This process can, for the purposes of this thesis, best be situated within the model of legitimate peripheral participation, especially as developed by Lave and Wenger.

This model is “a way of acting in the world which takes place under widely varying conditions” (Lave and Wenger 1991: 24). Such conditions in this ethnography would be found in the physical space of and social expectations about the lab. Learning thus happens not in “the acquisition of structure, but in the increased access of learners to participating roles in expert performances” (Lave and Wenger 1991: 17). Such a formulation neatly applies to the lab space – CS1, CS2, and CS3 students develop the appropriate skillset, an ability to problem solve in the
Computer Science context, everything that is needed to complete the requisite courses, all through a space that enables student proximity to technology and the peer advice needed to facilitate the process. Even if CS students do not use the lab space, their involvement in classrooms and with problem sets introduces them to a complex world, both inside and beyond the screen, which expands outward from Silicon Valley in waves of influence. Through the development and deployment of these participating roles, the students learn what modes of problem solving are appropriate and expected in the coding space.

Computer Science students are not, however, expected to mimic those in the “middle” of this increasingly globalized network. Instead, they learn how to perform in ways that reproduces what has been previously established (Lave and Wenger 1991: 16). “Teaching students how to think” implicitly means “teaching students to think how we (computer scientists and coders) think.” This is accomplished through the maintenance of structures of learning which allow for this reproduction to happen (Lave and Wenger 1991:16). Tutor services act to maintain the processual structure – they reaffirm and guide students to accomplish the set goal (e.g., finishing the problem set). The problem sets themselves also provide another iteration of structure. Finally, having to situate new-coming students into this frame in order to “think right” and thus be a successful student implies that belonging is not only a “crucial condition” for learning, but also a “constitutive element” of the whole epistemological process (Lave and Wenger 1991: 35).

Without structured belonging, learning cannot be achieved. Without learning, a certain kind of knowledge cannot be communicated or understood. As we will explore below, this has great consequences for those who routinely use the lab as part of their training in Computer Science.

After learning how to “think correctly,” some students find great freedom in their work within the department. Billy remarked that, after learning the rules, the possibilities were endless.
“It’s like the Department gives you Legos,” he explained with a smile on his face. “After a bit, you have a toolbox from which you can build anything.” In other words, the (coded) world is yours – provided you get through the initial acclimation process and know how to use the building blocks you have been given. Here, we begin to see the beginnings of the Bourdieuian “habitus” within the lab, a habitus that deals with how students should learn the material. By beginning at the periphery and working inward, to the coveted position at the center of the field, students learn to behave and think in ways considered “proper” to achieve success in their coding endeavors.

This habituated model from Bourdieu, coupled with the more specific model of legitimate peripheral participation, enables me to argue that knowledge production in the lab is an eminently social act. To do this, I draw most directly upon Jasanoff:

Knowledge and its material embodiments are at once products of social work and constitutive of forms of social life… Scientific knowledge, in particular, is not a transcendent mirror of reality. It both embeds and it embedded in social practices, identities, norms, conventions, discourses, instruments, and institutions – in short, in all the building blocks of what we term the social. The same can be said even more forcefully of technology (2004: 2-3).

Jasanoff goes on in later parts of her work to discuss how the model of the “co-production” of science and social order allows one to evaluate how material objects can shape and sustain “relations of authority” (2004: 4). It is precisely these relations that I explore, in greater detail, in this chapter. Grounded in what is being materially produced in the space, I argue that the social interactions that emerge between the students in the lab are heavily dependent on perceptions of authority, which creates a hierarchy among the CS students. This authority is what used to be competency in the space, transmuted into its new form along the lines of the theoretical model posed by de Certeau (1984). Competency refers to that regarding one’s ability to produce and
builds on the notion of the students as disciplined within the framework of the space. This
discipline is mediated, as we shall immediately see, through the screen.

**Screens On, Screened In**

Billy, a senior, is building a program that will be able to predict the emotions of tweets.
In his words, it will be like a neural network, trained on a database of about 40,000 tweets to
understand the nuances of technology-aided human communication. He has a small smile on his
face as he explains this to me – it’s working, would I like a try? I say yes, suitably intrigued,
waiting to see what such a complex program – a senior capstone project, no less – would look
like. He beckons me over to the computer and pulls up a tiny window, just a single line and two
buttons. It reminds me of one of the pop-up windows that comes onto my own screens when
some internal process has encountered an error. As I am prompted, I type in the first thing that
comes to mind: “looking forward to the weekend.” I hit Enter, and a small smiley face icon pops
up next to my “tweet.”

“Is that what you meant?” Billy asks, watching me carefully.

I nod, trying to wrap my head around the possibility of a program picking up on my
intended mood from just a few neutral phrases.

“It only knows two emotions, now. We started with 30, but our error rate decreased
when we used fewer. So it just knows positive and negative with about a 75% accuracy rate,
which is on par with human perception” Billy says, moving his chair to its original position,
squarely facing the screen, as I step aside. “And sometimes I can trick it.”

I ask him what he means by “tricking” it.

“It gets confused when you use swear words, and will say your message is negative when
it’s actually positive.” Another smile, as if to imply that he enjoys it when his code does not
accurately work, as if the point of creating this “neural network” is to somehow still prove competence over the machine. Perhaps he means it as a testament to the complexities of human language, crafted from the ultimate, most organically perfect neural network – our brains.

I thank him for letting me work with it and remark that it reminds me of *Westworld,* which one of my close friends had recently convinced me to start watching.

He beams. “I love that show!”

It *is* a good show, albeit eerie in its suggestive possibilities. I do not let on that I am relieved that semi-sentient technological programs that could read – or, as in the case of *Westworld,* feel – complex human emotions are not yet being created with success in the F&M Computer Science lab. Instead, we let the success of the moment, of the semester, settle quietly over where we sat in the empty lab.

The amount of work that was put into that project was surely astronomical; the only part of the process to which I was privy was the result. This interview was held at the tail end of the semester, after all the weeks that were spent assembling it. And, as I mentioned, that result was a tiny window, almost anticlimactic in its grey, utilitarian and functional appearance.

What Billy had actually asked me to do was to handle the carefully designed user interface, the culmination of lines and lines of code. It was simple, automatic, for me to put in my tweet, to press the buttons and see the cutesy icons. The notion of me reading the code was never brought up in our interaction – I did not even think to ask, in the moment. In this way, a dichotomy was upheld: I, the user, see this one side, this tiny grey window, if only because it is what I expect to see. Billy, who wrote the code, is the only one privy to its nuances – it is *his* in that it is not shared. If he were in the workforce already, it would be “proprietary code.” Only he sees what is behind the grey box. It is his hand that did the calculations and wrote the lines on a
separate, hidden window, implicit within the external appearance of the box, but not referred to. I am only the external user. He is the internal supervisor, one who can also access, see, and understand its external component (and who can do so with a greater comprehension than I could achieve in that brief interaction). He is the external puppeteer who constructed the processes that determine the functionality of internal understanding, the mediator between external and internal. The point of mediation, here, is the screen.

Through learning to write and interpret code, students are given the keys to a realm few can knowingly reach. They are allowed behind the screen and into a land that is predominated by many different languages, algorithms, and possibilities. This access makes them into “coders,” distinct from any “user.” Coders code – they produce an object. This object can be consumed and has a perceivably intrinsic value that stems from its (Marxian) use-value. The process of coding also reaffirms the identity of “coders.” In the most Marxian sense of the concept, their personhood as computer scientists is built through the process of producing these products, which then interact with other social people (Marx 1978: 23). The social relations which stem from this production are unique to computer scientists – thus, to learn how to write and read code is to be inculcated into a specialized group of producers.

Additionally, this form of code production – and its subsequent mystification – is only made possible through the involvement of screens – it is the screen that displays the product and allows for others to engage with it, forming a perceived connection between the coder and user. A divide is also erected, in which the user cannot access the material behind the program. To harken back to the previous chapter, the screen allows that which is produced in digital space to be consumed, but it can also prevent other, non-qualified users from producing within that space. The physical code, in its gory, messy detail, is untouchable to all except those who have the tools
or knowledge with which to access it. For the purpose of this argument, a user is someone who uses computer-dependent programs but does not have the knowledge or skills to create or manipulate these programs. A coder, on the other hand, can also be a user, but the key distinction is that the coder is also capable of producing computer programs for use by others.

In this sense, any interaction between a user and a program is an interaction between the external world and the internal, ethereal world that is contained behind a screen. Furthermore, since the internal, digital space is both the culmination and product of coder production, and this production is the condensation of the labor and therefore of the persons of these coders, the interaction between user and program is also the interaction between user and coder. This is not obvious from the interaction – while using Billy’s program, I could not tell that it was Billy himself who constructed it. Similarly, his code could not tell that I was the one manipulating it – just that it was being manipulated. The screen thus becomes the place at which these two worlds meet in a more masked manner. Much like how Billy’s knowledge has been masked as object code, the coder must be able to – as if magic – transform that which they knew into that which can act. This requires knowing how to retreat behind the screen. These actions are then processed as product, sometimes with utter passivity (i.e. scrolling through a social media platform when bored), by the user. Through knowing how to retreat (and therefore how to produce), a user is transformed into a coder. This is the culmination of the discipline discussed in the past chapter – all exercises prescribed by the Professor are meant to enable and train students to produce utilizable code.

Such mediation – being inside or outside the screen, or being in both spaces simultaneously– is implicitly acknowledged by many students in the department. For these students, there are no rituals, soundtracks, or moments that best abet the flow of knowledge from
the abstract form inside them to the more concretized windows on the screen. Instead, coding is
an all-consuming task that can be done in any moment – all one needs is a screen (and sometimes
a keyboard). Meredith, a CS1 student, remarked that she could be at a concert and she could still
be coding because of how “into it” she gets. This sentiment is echoed among other students –
when it comes time to code, everything else fades away. Coding can be done anytime, anywhere.
Meredith further specified that if she had a computer at this hypothetical concert, this retreat into
the coding world behind the screen would be even more possible. The retreat is perceived by
Computer Science students to be seamless and all-or-none, not dependent on any external factors
besides having access to a screen. Nor is this a sentiment that is solely expressed within the
population of more advanced students: as noted above, Meredith was a first semester computer
science student at the time of this interview. Billy, the student with the complex senior capstone
that began this section, also mentioned experiencing this phenomenon.

However, by expressing how “into it” they get, the students are also communicating
another key relation: by coding, the students get “out of” everything besides coding. To students,
coding therefore takes form as a moment in which they retreat from their “external” lives to
some other place, a place that can only be accessed through a screen – a place behind a screen.
The screen is a mediator between awareness directed outside the screen and that directed within,
which, furthermore, depends on focus. As long as the code and the work captures the focus of the
student, the student’s consciousness will reside behind the screen. If a student’s focus is diverted
to, say, a friend greeting him/her and sitting next to him/her, or by his/her hunger or need for
sleep, his/her attention is drawn out of the screen and into the external world.

The screen may thus be the point of mediation, but the focus of the student’s
consciousness is the fulcrum on which this attention is directed. This moving focus allows a
student like Billy to spend hours within his code, poking at its edges and unraveling its details, only to later interact with it from the more simplistic, fool-proof outside perspective, tapping a few keys and evoking either a smile or a frown from a simple box. In the latter case, his focus was on the program’s appeal to the consumer/user – “Is that what you meant?” If anything, he was focusing on my interaction with his program. Did the signifier adequately represent that signified?

The focused consciousness of the student then structures the interaction with the program. If the focus is internal, the interaction is intimate, nuanced. If the focus is external, it becomes as automatic as typing a line into a box. Such online intimacies are learned – disciplined – from the very beginning, in CS1, as students are learning how to produce their own codes. In the anecdote opening this chapter, Allie, the CS1 tutor, wrote the bit of the student’s code on the board, turned to her, and said, “Okay, you’re in the ‘while’ loop. Where do you go from here?” By phrasing the question as such, she was prompting the student to retreat into the code, to truly understand what was happening. Without this intimacy, understanding and therefore production cannot be accomplished by the student. Moving behind the screen is a concept that is built into the very foundation of the discipline, and it is learned as soon as students begin the process of writing their own codes. Therefore, learning how to move behind the screen – to screen oneself into this all-consuming state of production – is learning how to code. While the screen acts as a metaphor, the code is concrete, only changing by the hands of those with the knowledge and access to do so.

Despite this ability to switch between inside and out, once the internal computing space is reached and code is produced, the student cannot fully re-engage with the external world. Something always remains within, interwoven between each line and break. Reaching intimate
understanding with the code and the process of coding requires the sharing of intimacy of the coder with his/her product, or code – this is the opposite of alienation from one’s product (Marx 1983: 141). As the code becomes easier for the student to understand and manipulate, more of the student is left behind as labor and time given to the program (Marx 1983: 140). It is from this intimate work, then, that users can interact with the program and, by extension, with the coder. The user, by using the code, is in effect interacting with the pieces and remnants of the coder’s (codified) knowledge. Learning how to code is learning how to share intimacy with others. To do so, though, a coder must be able to be fully intimate with his/her own code.

This intimacy is a form of knowledge – code as a product is the condensation of the coder’s knowledge into accessible and interactional effect. This is most directly a condensation of intangible abstraction – human knowledge – to physical product, code. This physical product can then be manipulated by others who consume it. Materialist thinking supports the notion that a product is made into a product through its consumption (Marx 1978: 229) – therefore, coding is the condensation and manipulation of specialized knowledge for consumption by others. If users such as me do not use or interact with programs such as Billy’s, Billy qua coder becomes irrelevant. The use-value depends on the function of the code – if this factor is not present, the consumer will have no need for the code, and the coder thus becomes irrelevant. This user-coder interaction is tantamount to the coding process, which means that the screen, as the mediator, takes on a hugely important role in the construction of a coder’s identity as a worker.

The identity of a coder is not just that of a knowledge producer – knowledge is additionally situated within a social context, so the coder becomes a reproducer of social knowledge. Society can be ordered within the frameworks established with technological innovation (Marcuse 1982: 139). To do so, coders need to give their product social salience – not
only did Billy construct the material code, he had to couple it with a process of signification so that his code could be understood in a more colloquial, accessible manner. He had to create a culturally understandable context for his product. My tweet elicited a smiley face, which provided a signifier informing me of the result of the program. As a user of his code, I was able to understand the product I received from my own input (“Looking forward to the weekend”) through this signifier – thus, my understanding of the meaning of my own words was minutely altered by the signifier, giving the product a place in my own social understanding. Through this joint process of fabrication and signification, a coder translates abstract forms of knowledge and converts these into a social context; through this process, the object (code) is defined and “rationalized” (i.e. within the pre-existing social framework) (de Certeau 1984: 20). Not only does this process yield and/or reinforce the types of “normative frameworks” with which de Certeau is chiefly interested, it creates understanding on the part of the user (de Certeau 1984: 21). Interactions with screens then create moments in which social frameworks and the understanding thereof is created and reproduced, both within the screen on the part of the coder and outside, with the user. Coding, then, cannot possibly be removed from a social context, since sociality is necessary in contextualizing the code and providing its relevance. Without the signifier of the smiley face, I would not have been able to understand the code. To return to Jasanoff’s argument – society and knowledge are co-produced (2004: 2-3).

However, I as user was also compelled to accept a proposed framework that did not quite fit my perception. My neutral message – “can’t wait for the weekend” – would not merit a full, beaming smile if I were to share that with a friend. By constraining a plethora of messages within the realm of either “positive” or “negative,” Billy is implicitly asking that his users make slight compromises within their more nuanced understandings of emotion and language to fit within
the framework of his program. Through this compromise based on necessarily altering one’s understanding of social phenomena, Billy alters the way in which one thinks about these social phenomena – and here, his production takes on a social significance based on his own understanding of the code. He was concerned with correctness of perception, which required a sacrifice in specificity. My positive feedback – “yes, I meant that” – reaffirmed this sacrifice. Did I truly mean that? I did not mean my message to be negative, so sure – but acknowledging this required me to gain a new, more inclusive understanding of “positive.”

Coding, then, has a social context outside of the screen. This social context further reproduces itself through the code’s use by users. The users have to accept the product and use it in order for the product to influence the society of the users. (This, as we will explore in the next chapter, has implications for how welcome women feel in a heavily masculine field – accepting the masculinity of the interfaces being generated is often difficult for female users and reaffirms the notion that computing is for men.) In addition, though, coding is influenced by a more internal social context, which also reproduces itself. This context encompasses power structures, which must be reproduced as well especially because there is only one space for code to go–behind the screen, in the digitized space shared by all who use it. This sociality takes on a hierarchical form, organized based on perceived knowledge within the specialized group of coders. This is especially so when considering the role technology plays in organizing social structure, as Marcuse does when he states: “Men, in following their own reason, follow those who put their reason to profitable use” (1982: 148). This becomes most clear when considering peer groups within the lab, the largest and most developed of which is the CS3 student group.

You’re Hardcoding?
We now return to the group of CS3 students sitting in their circle, coding – a noisy process that belies any vision of the coding process being one of solitary detachment. They were working on the same assignment, but doing so on their own terms, on their own screens – as mentioned above, plagiarism is not condoned within the department. As they worked, they joked: What language do coders at Hogwarts use? The answer is “Parsertongue” (since coders “parse,” this is a play on Parseltongue). That also makes them fluent in Python, eh?, someone else chimed, continuing the joke with his reference to a coding program. Everyone chuckled, as one must when confronted with a pun such as that. Occasionally, the chat turned to the assignment. Someone mentioned a quiz to another, who quickly scoffed and said that he got the answers from Alec, implying that the person with the question should ask Alec.

All eyes turned to Alec. He shrugged, then claimed that he had to watch an associated video to figure out what was going on. This seemed to mollify the group, and they all returned to their respective assignments and chatter. Every now and then, someone would direct a question to Eric or Alec, both sitting in the front, calling out the man’s name to catch his attention before following up with the question.

Eric, sitting next to Alec, caught a glimpse of his screen and exclaimed, “Dude, you’re hardcoding?” Everyone else paused to show surprise as Alec sheepishly admitted to engaging in the process. Alec began to clarify the process to Eric, saying that he decided to use vectors to obtain the desired result.

“Ah, smart. That’s almost what I did,” Eric said, turning back to his own code at that point.

Alec nodded at the recognition, also turning back to the assignment at hand, reading over the requirements once more. He paused for a moment before stating, “I’m just going to jump
right into it… you’re actually going to get out a sheet of paper and plan it out?” He was referring to advice given by the professor for the assignment. The use of “you” was hypothetical, as if he were talking to the professor. The professor, however, was obviously not present – but the challenge was there nonetheless, hanging in the air without a reply.

One guy in the group, sitting a couple of rows behind Alec, was having trouble with the assignment. He asked a question to the group – from his tone, it was clear that he was confused. All the other guys laughed at the question, and one said, “How are you a CS3 student and not know how to compile your code?” The guy shrugged, laughing as well like he understood the joke.

Jack entered, and everyone tossed greetings his way. He strolled around the circle, making a point to greet everyone with a casual handshake and address the couple of questions posed to him. His first response was to shrug and tell the group that he “did all of this basically from scratch,” the delivery matter-of-fact. As he walked behind the guy having trouble, he laughed. “What the fuck kind of error is that?” He asked, shaking his head. “I have never seen that before.” The guy with the error cracked a smile, looking self-conscious at being called out, before admitting that he was not sure how he got it. But Jack was already gone, moving on, waving goodbye as he exited as quickly as he came in.

Moments such as these become instrumental when considering social structure among peers. This was a Wednesday night, with no tutor present, so the CS3 students were compelled to turn to each other for help with the assignments. What is most notable is how the students called upon certain people for help. Eric and Alec, and occasionally Dave, were the ones being asked the questions, until Jack walked in and all the questions were directed toward him.
Here, we see the creation of authority at work. By this phrase, I mean that what a certain person says or does is more respected than someone else’s words or actions are. Such a creation seems to be based on perceived knowledge. Alec is situated in the discursive space of the CS lab as someone who has the answers for the quiz, thus everyone considers him the authority on the assignment. This authority is represented by the imperative, directed flow of questions – “Ask Alec, he knows.” This locution implies that only Alec is to be trusted with the answers on this particular subject.

In this system of authority there are established frameworks by which an “expert” becomes someone, within a specialized field, who is perceived to be knowledgeable – evidenced by the competence with which [s]he can “mediate between society and a body of knowledge” and which becomes exchanged for authority (de Certeau 1984: 6). Such discursive exchange between peers using the lab is a process whereby competence is “transmuted” into “social authority” (de Certeau 1984: 7). In the case of the lab, this type of exchange happens when a particular student is nominated to answer a question, or translate a concept and make it accessible to the rest of the “society” within the lab space. Without this nomination by a peer (or the associated publicity of being asked a question in the normally quiet lab space), a student could be deemed by his/her peers to be competent with the material yet not authoritative on the subject. Being nominated to answer a question then allows the student to rise to the occasion and answer the question—thereby answering the challenge of competence that had been issued. The ability to answer the question posed reaffirms the student’s competence in the eyes of those using the lab. Of course, a student could be nominated his/her CS lab peers, yet choose not to respond to the nomination. The student may then be seen as competent yet not authoritative, or neither competent nor authoritative. Regardless, without a moment of nomination, this exchange
has no bearing and does not happen. Students who offer comments without being asked are not usually heeded, and those comments thus do not form part of the social knowledge of the lab space.

This form of nomination is a moment in which authority is generated interactionally (Benoit-Barné and Cooren 2009: 7). Authority is thus bestowed through the consent of the collective – it does not originate from one individual (Benoit-Barné and Cooren 2009: 9). In the lab, this consent is given when one asks another for his/her advice or knowledge. One person standing up and announcing the answers to a given problem set would not generate the same authority – it is not the answers that matter so much as it is the interaction that takes place between the students. Creating this authority also serves to communicate something to the rest of the people using the space – interactions in which one person is asked to help another emphasizes that being a competent coder is how one is noticed and given authority in the space (Benoit-Barné and Cooren 2009: 12). It is along these interactional lines that competence is transmuted into authority.

When competence is exchanged for authority, de Certeau recognizes that there could be a moment in which the expert is perceived to have less competence, since it has been converted to that authoritative form. A paradox then develops in which an expert, who may have less real competence or knowledge nonetheless is granted more authority than he/she, perhaps, deserves (de Certeau 1984: 8). This is not an intuitive process – the expert is not aware of this exchange. Instead, the expert will see the authority as something gained with (not in place of) competence, and begin to apply this authority in other areas which might demand competence. This creates a moment in which authority will begin to expand, to transgress (spatially) into other places. Just as this spatial transgression will leak further out (proportional to the authority the expert
perceives himself to have), there needs to be a center for this leak. The expert, then, will stake his authority on a spatially marked place to note from whence his authority stems, which is the husk of his former competency (de Certeau 1984: 8).

In the case above, this is what is happening with Jack. As he walked into the lab, everyone turned his/her questions to him. Jack is no longer a CS3 student – he used to be, but now he has moved beyond that level to take elective classes exclusively. Jack is therefore perceived to have been a successful CS3 student. His competence in CS3 materials was realized through his performance in the class, which secured him an exceptional grade. This competence, moreover, is directly related to the discipline that has been instilled in him through the various exercises with which he has been engaged over the past 3-4 years. This tested competence is now exchanged for authority in the lab space through interactions between himself and younger students. Jack is still perceived by the CS Lab denizens to be an authority in CS3 material, so he is treated as such when he appears. When faced with questions, though, his answers are not specific: “I did this basically from scratch” is a non-answer. A non-answer, but one that asserts his authority (and therefore power) within the lab and in relation to those listening – the space from which his authority stems is that completed assignment. He had successfully completed it in his time in the class, even if he cannot remember the specifics. Through moving beyond the material in CS3 and trading his competence for the authority that comes with successfully forging ahead in the program, Jack is no longer competent enough to ask for specific details on the subject.

The discipline of the students, discussed in the previous chapter, helps establish society as the “objective entity” of Marcuse’s theoretical model (1982: 159). As students learn to orient
themselves within the space, society becomes set, normalized as rules and behavior that need to be followed. More importantly, though, it is:

…noticed chiefly as a power of restraint and control, providing the framework which integrates the goals, faculties and aspirations of men. It is this power which collectivism retains in its picture of society, thus perpetuating the rule of things and men over men (Marcuse 1982: 159).

Society, as a force that demands disciplined adherence, is indeed providing restraint – to stray outside of “normal” behavior is to stray outside of accepted social norms and is noticed. Collectivism – which Marcuse identifies as existing so long as the “individual interests are antagonistic to and competing with each other for a share in the social wealth” – necessitates a hierarchy, chiefly due to its competitive edge (Marcuse 1982: 159). The point of competition is to have a winner– the winner, [s]he who is most effective in his/her use of the space, and thus more competent in his/her production within the digital space, is then perceived in relation to those who have not attained his/her success. This hypothetical winner is then awarded the authority that replaces his/her competency.

However, when Jack uses his power (conflated with authority) without the competency to support it, it could be considered an “abuse of power” (de Certeau 1984: 8). This also marks a moment in which authority within the lab becomes a transcendence above other forms of authority in the space. “I did it from scratch,” Jack claims – this statement aligns closely with Alec’s “You’re actually going to get out a piece of paper and plan it out?” Establishing one’s own authority and power within the lab means not aligning with the most visible point of authority within the Department: the professors themselves. The authoritative students base their authority off a perceived transgression of established boundaries of authority.

One way in which this moment of transcendence is made apparent is through the prevalence of joking among the students. In his ethnography on relationships between collegiate
fraternity brothers (the “fraternal bond,” as it were), Lyman identifies joking, especially sexist joking, to be a mechanism by which group solidarity is maintained and internal tension dissipated (1987: 150). Through these jokes, men were perceived as attempting to project a “cool” façade within the rules of society while expressing emotions and tensions that might damage the framework within which they operate (Lyman 1987: 159). Anger is thus controlled, restrained, through cursing or aggressive ribbing – Jack’s, “what the fuck kind of error is that?” or Dave’s, “How can you be a CS3 student and not know how to compile code?” Most striking was Lyman’s perception of joking as a response to a fear of being dependent on or subjugated to a form of authority outside the men’s own – many of these jokes involved women (who the fraternity brothers were dependent on for their own sexual desires) and authority figures such as their own parents (who act as physical reminders that these men will one day have to be subject to the same drudgery of the workforce as they were) (1987: 157). Thankfully no vulgar, sexist jokes were being traded in the lab – but there was a plethora of jokes mocking the expectations of the professors as well as those who did not understand the material. This, to extend Lyman’s research, could be externalized, controlled responses to the stress students face from their assignments and, more overtly, from the expectations their professors might hold for them. Projecting a “cool” façade, as it were, is one way of seeming authoritative in the face of academic stress.

Joking aside, students still persist in asking Jack their CS3 questions, despite the vague answers they may receive. Even though – or perhaps because – authority is based off competence – yet inversely related to competence – it is craved. Students want to have authority within in the lab, because authority has been related to success. Those who have authority had competence. They were able to produce that which could be exchanged for a good grade in the
class, which can then be placed on a resume or taken into account by future employers – in the academic system, to attain a good grade point average is to attain success.

Success within the lab is then associated with academic success within the department. This academic success, grounded in one’s ability to produce, is publicly recognized and thereby traded for authority, and this then creates a gradient comparing peers within a group to one another based on their authority. Here, a hierarchy is established where the students at the top have the most authority, compared to those below, who have less authority. In the example above, though, what becomes apparent is that the students with less authority become those who are nominating the other students and recognizing their accomplishments by directing questions to them. Alternately, students with more authority can reinforce this hierarchy by displaying their authority to those who are not as competent. By Jack telling the one student that he had never before seen that error before, he was leveraging his authority to imply that, in all his time as a competent coder, he had not created such a buggy code. Therefore, this student is not at Jack’s level – he is not competent enough, yet.

Students can also defend their competence within this structure, as seen in the interaction between Eric and Alec. Phrases such as “that’s smart” – a recognition of competence and therefore a grant of authority – are sometimes followed by a justification on the part of the nominator. In the case of Eric: “I almost did that, too.” By including this, he is showing that he too had the competence to think of this method that is considered “smart,” so he should also be considered competent, in case the day comes for a nomination moment in which he could trade this competence for authority and a higher status within the lab. After all, successful students are considered successful coders, and successful coders get top jobs at Google, Apple, Microsoft, etc. Success is then inextricably linked to the image of a coder – to become a coder, a student
must succeed in the Computer Science department by doing well in the required classes. This, in turn, must be recognized by some verbal acknowledgment, either on the part of the student or by someone else.

Experts are also chosen by a more external source, that being the department itself, detached from yet dictating every part of the lab space. Professors, not a part of the after-hours lab space, will nominate students based on performance within class to be tutors for others. Lab teacher assistants (TAs) are included in this nomination process. These tutors will host official drop-in hours and sit in on classes (or lab sessions, in the case of the TA) to answer questions or refresh their competence with the material. This is a more formal, professor-driven nomination of authoritative students which leads these students to act slightly differently within the lab. A few will sit at the lectern while they are holding their drop-in hours, as if to represent the professor in the professor’s absence. Additionally, some students will go to the lab after hours explicitly because the tutor will be there to answer questions. However, tutors are still called over, still questioned and probed for knowledge. They are still perceived as successful, and their competence is still traded for authority.

This became clear to me when I observed an interaction between Allie and Luke, a tutor and a student who are the same age. Sadie had called Allie over to her desktop to ask a question, and the two were talking when Luke crossed the room past where they stood, ostensibly to leave the lab. Sadie waved Luke over to show him her code, which thanks to Allie was now working. After sharing her success, she asked him about the Rubik’s cube he was holding in his hand.

Could he really solve a Rubik’s cube?

Luke: yeah
Sadie: That’s pretty cool. My friend was able to do it with his eyes closed and I think that’s wild.
Allie: (2.0) Like, I know I could learn how if I just watched enough YouTube videos and memorized =
[ ] [ ]
Sadie: There’s definitely a thing yeah
Allie: =whatever pattern it is that they know how to do real fast.
Sadie: It'd be a fun party trick, but o(h)ther than that that's a lot of effort to put into it =

[ ]
Allie: ((laughs))
Allie: = right
Sadie: definitely
Ellen: oh man
Allie: Although, I did see a video where it's li:ike teaching you how to cheat learn how to solve a Rubik’s cube because there are some, like, loops where you just do like this move this move this move over and over and over again you'll get back to the same part where you were before so if you take

[ ]
Sadie: yeah=
Allie: =a solved Rubik’s cube and you go like halfway through the sequence it looks messed up
Sadie: yeah =
Allie: =and then you can just take it and finish it ((laughs)) a(h)nd like show people like ((purposefully high pitched)) oh yeah I can solve a Rubik’s cube. ((laughs))

[ ]
Sadie: but it's really just
Ellen: huh
Allie: ((laugh))

Luke’s competence with the Rubik’s cube, when verbally acknowledged by Sadie, prompted Allie to justify her lack of competence when faced with a perceived challenge to authority. By justifying that she could learn how to solve a Rubik’s cube like Luke, not only does Allie take the moment to talk up her competence with coding programs and patterns through comparing the puzzle to a code, she also twists any perceived competence held by Luke. If any good coder can solve a Rubik’s cube, then that does not make Luke’s fun skill all that noteworthy and special. However, this can be perceived as an extension of authority based on place (her status as a tutor) which could be considered an abuse of power (de Certeau 1984: 8). By logically making the connection between being a successful coder and being able to solve a Rubik’s cube, Allie is extending herself to retain authority in the eyes of those watching. The “abuse” here comes from preventing Sadie from fully establishing Luke as authoritative in the space.
Students also defend their competency frequently when interacting with tutors. Often, students seeking help from tutors will begin their questions with what they know. For example, a student might say that (s)he has tried a certain approach before realizing that it was just making his/her code buggy – what should (s)he do instead? Additionally, these justification processes might appear after the student receives an answer from the tutor. “That makes sense,” the student might say, then show his/her competency with the material by explaining the conceptual reasoning behind the proposed solution, even if the student was not prompted by the tutor for this explanation. Through this process, the student is careful to ensure that competency is not sacrificed through eliciting help from the authority, much like how Eric included his justification after praising Alec’s approach to the code.

Acknowledgement processes also take place between non-peer groups. For example, students in CS1 and CS2 can frame CS3 students as competent through their interactions with them. Like the interaction seen between Jack and the CS3 students, this can take the form of younger students asking older students for help on assignments the older students completed in the past. Other times, these interactions do not explicitly reference assignments, but take the form of greetings.

“Welcome, boys!”

To enter the computer lab is to be immediately noticed. Even if one is not seen at first, one is heard – one must walk up it, the hollow clunks of one’s shoes on the structure alerting everyone to one’s entrance. Upon my first entrance, I found this mildly intimidating – everyone immediately, reflexively looked up at me as if to gauge my presence, then in that same reflexive moment returned to their attention to their separate screens. The only one who remained attentive
to me was the student with whom I was due to meet, who smiled, said, “Hey Ellen!” as if to alert
my attention to her, and waved me over to sit next to her.

This initial greeting has many variations, yet the greeting as it plays out within the lab is
rife with authorization and recognition processes, much like what was explored above. The most
immediate relation that is created is that between a less advanced student and a more advanced,
usually older student. Additionally, such speech acts are usually implicitly gendered and uphold
methods of discourse that work to exclude women by situating them outside the realm of the
average, male Computer Science student.

Dave – an upperclassman who has been taking Computer Science courses since the first
semester of his freshman year – entered the lab. Immediately, the one male CS1 student sat
upright and called out, “Hey man!,” which effectively caught Dave’s attention. At the freshman’s
request, Dave sat down next to him and began talking to him for a few minutes, before turning
around and chatting with me; we had lived on the same hall freshman year, so I took the
opportunity to interview him. After our talk, he swiveled his chair so he was once again facing
the two freshmen, and as I was leaving, he was helping the first female student, the one having
some trouble at the beginning of the night. Dave was becoming the “Jack” among the CS1
students, displaying his authority when asked to do so.

The greeting can be a tool utilized by clever underclassmen seeking help on assignments
while the tutor is otherwise busy. The pattern of a younger student calling out to an older student
upon that student’s entry into the lab is mirrored among CS1, CS2, and CS3 students (who are
usually calling out to a select few individuals who have almost completed the major
requirements in their entirety, like Jack). By understanding the goal of receiving help, the
greeting then becomes a method through which underclassmen recognize certain individuals as
capable of giving help – these individuals are the experts, their competency traded for authority. Follow-ups to the greeting usually include questions such as “what did you do for this assignment?” or “does my code look alright?”

An additional component to this pattern is that it is just the younger students calling out. In my time in the lab, I did not witness a single instance in which an older student called out to a younger one. This demonstrates that the greeting is an authorization process through which less experienced students reaffirm the perception of knowledge held by their older counterparts, usually to leverage some of that knowledge for their own personal gain. Competency is recognized, and thus traded for authority as the experts comply. Older students are noteworthy and exceptional additions to the lab space precisely because they can aid another student in becoming more knowledgeable. For example, seniors Lily, Billy, and Jack all experience warm greetings from the rest of the students, mostly because one is a tutor, the other is a lab teaching assistant (TA), and the other is always able to help those who ask. Tutors were not the only students being asked for help, however, they did make up about half of the students who were most called upon. This implies that recognition and the perception of knowledge among younger students does not exactly align with the professors’ choices of knowledgeable students. Students have created their own hierarchy, which is based on authority, which then traces back to perceived competency. Displaying knowledge of this social hierarchy then becomes a moment of competency in itself.

Furthermore, the fact that the perception of an individual’s knowledge is associated with the greeting he or she may receive creates and reproduces status which then maintains that social hierarchy among those who use the lab. Those who are perceived to be the most knowledgeable are consequently those accorded the highest status within the space. By this, I mean everyone
wants to hear from them, sit near them, do what they did, in hopes of one day becoming that person (or at the very least, in the hopes of receiving a passable grade on the assignment). The demand for their time and wisdom is high, and these high-status students perpetuate that demand by acknowledging the greetings and answering the students’ questions. Sometimes, these older students are not even using the space to study, themselves, but as a study break in which they stop by the lab to see who is in that night. Through this, these older students are purposefully seeking out a social interaction from others in the space. They are also reproducing their authority and status with each appearance – if they never show up to the lab, who would remember their authority?

Not only is the greeting a moment in which a knowledge-constituted hierarchy is established, it is also a process of recognition. By engaging with the older students, the younger students are ensuring that they are recognizable and active members of the lab. This then could lead to more help on assignments or a friend in the room, but it could also be feeding into the previously described hierarchy. This strategy, in which younger students recognize the accomplishments of the older students and thereby reproduce the older students’ authority (causing the leak mentioned above), is not a new phenomenon. This same strategy is seen in the business world, which makes sense considering that the field of Computer Science is essentially a field of entrepreneurs and businessmen. Succession in managerial positions is heavily dependent on visibility – those who are “visible in the right way” are those who are chosen to succeed those out the door (Holgersson 2013: 62). While seniors in the lab are not always consciously choosing those who will succeed them as “expert” within the Computer Science department, being friendly with them does improve the chances of obtaining help from them on
assignments that the upperclassmen had successfully completed in the past. Help from the expert generates competency on the part of the helped, which then generates authority.

To gain this competency, though, the students seeking help need to know how to ask. At this point, differences in discourses begin to splinter the group into those who are able to secure help and stay, and those who cannot get help or recognition and thus do not feel welcome within the space. While Sadie called out to me, it was because she knew I would be meeting her there, not because she knew I could help her with her code. That instance was an anomaly, in the grand scheme of the lab, if only because my presence was itself an anomaly within the lab. Usually, it is the men who are calling out to or being called upon – Lily is the only female upperclassman who was called out to in this manner. The female CS1 students did not call out to anyone, male or female, who walked in, preferring solely to utilize the services of the tutor, and the other women who frequently used the lab space, such as Sadie and Holly, who are CS2 and CS3 students (respectively) were not called out to at all. The greeting, then, is a male homosocial phenomenon – it becomes defined through and by a “logic of a patriarchal system” (Connell & Messerschmidt 2005: 832). Those who begin to get excluded, then, are the women in the room.

Of course, male peer relations are also strengthened through greetings. Some of the students vocally welcome their friends who enter the room. For example, two male students walked in together one night, talking, and immediately spotted a friend of theirs, who had previously been sitting by himself. This friend leaned back in his chair, opened his arms, and said, “Welcome, boys!” before shaking each of his friends’ hands and gesturing at them to sit next to him. This was more casual than the greeting given to upperclassmen, more akin to the greeting that Sadie gave me. Sometimes, a greeting is just a greeting. But mostly, the greeting is a process through which homosocial male relations are created and maintained in the lab.
“Welcome boys” – we have now entered a space that couples a hierarchy based on authority with gender. We are now in the “boy’s club” of computer science – and we can now see how authoritative processes and differences in discourses truly are able to play out in the lab.

Gender in the Lab

The gendering of sociality within the lab was suggested during my very first interview with a student in the Computer Science Department, before I had even stepped foot into the lab. I was searching for a research focus for my ethnographic project and had decided to interview friends to pinpoint potential areas of interest. At the time, I was telling friends that I was
interested in moments in which they felt gendered on campus. As I sat down with Sadie and began talking to her, the scene that she described sounded outrageous to someone such as myself, who had never been exposed to the culture of the school’s Computer Science Department. “There’s actually this one time specifically last year,” she began,

I wrote… a pretty intricate piece of code, and like that worked and did something and so I was working with a [male] partner and I was asking what… like integer, string, like what kind of thing they wanted me to give it so that their program could like interact with it, and they proceeded to look at my code and… explain to me how to do it… and I was like, okay, so I wrote this, so I know how it works… Now please answer my question, so I can further help you.

While she called this moment part of a series of “microaggressions,” specifying she did not consider this to be part of a pattern of sexism (while never specifying what she did consider sexism to be), Sadie continued to recount another, less specific, moment in which she wanted to leave the lab after a male student made a comment to her. From these two anecdotes, I decided that it was necessary to consider the role that gender plays in Franklin & Marshall Computer Science.

Unfortunately, as it turns out, Sadie’s small narrative fits into a larger pattern of gender discrimination within the Computer Science field. As a young girl, I heard stories from my mother, a contracted software engineer employed by a national firm, of female coworkers who were harassed, were not taken seriously, or were simply missing. For example, a project characterized as having “mostly” women working on it in fact had just as many women working on it as men, and this project is usually one out of five being carried out by the branch at any time. Research concerning gender in the professional computing workforce has posited that women either voluntarily choose not to enter or remain in computing due to outside, perceived social pressures. Hayes raises the point that “evaluation biases” that favor men might be
prevalent in industries with fewer women (2010: 28). Since evaluation bias is the tendency to judge one’s abilities more positively when one matches a perceived stereotype within a discipline (Hayes 2010: 28), Hayes is implying that masculine traits may be an important factor which influences the perception of a computing professional’s abilities. These biases are historically situated, i.e. through metaphors which inculcate new ideas (such as computing) into society by comparing them to some preexisting cultural framework (van Oost 2000: 11). For example, comparing the computer to a brain – as was done in the 1950’s – is more subtly a comparison of the computer to prevailing masculine-typed abilities such as thinking. This then led to the public perception of a computing professional which fit in with that masculine image (van Oost 2000: 12-13).

This original perception is then perpetuated through heavy gender imbalance within the field, which reinforces the notion that computing is something men do (not women) (Peiris, Gregor, and Indigo 2000: 34). When most of the computer interfaces are being designed by men, femininity is not incorporated into the realm of computing – women then feel less inclined to take to the field since it comes across as heavily masculine. This cyclically perpetuates the issue, since this keeps the field male-dominated, allowing for the creation of new, male-oriented interfaces (Peiris, Gregor, and Indigo 2000: 35). The masculine image of computing is being continually perpetuated and marketed to users, which might deter women from developing an interest in the field. Ironically, many software companies recognize the deficit in the overall skill set of their workers that arises from this gender imbalance. Female workers typically have better interpersonal skills, are more able to work within teams, and are more motivated to produce useful product (instead of just playing with the code). This makes them more desirable employees (Abbate 2012: 77; Peiris, Gregor, and Indigo 2000: 37). However, Abbate mentions
that managers within companies are often selected due to their “macho,” competitive (so classically male) attitudes, which perpetuates the “egotistical” attitude of the industry, as a whole (2012: 77). Finally, masculinity is privileged in Western society – there may be a subconscious concern that programming might lose its prestige, should it reject the masculine egotistical style for one that is more inclusive and “egoless” (Abbate 2012: 78).

There is both an implicit and explicit, recognized need for women in the industry (Margolis & Fisher 2002: 2; Peiris, Gregor, and Indigo 2000: 37). To acknowledge this deficit, however, we must confront it in its explicit form, through the direct question: Why are there so few women within the industry? My research does not directly address industry concerns – however, Hayes identifies similar social pressures influence the gender disparity of the workforce and classroom, which makes research involving undergraduate education analogous to these concerns (Hayes 2010: 34). While considering ethnographic moments in the lab like the above, I understood that the lack of women in the field more generally was pertinent for my consideration of lab space at Franklin & Marshall College. As my understanding of sociality within the lab grew, so did the conviction that this sociality required an analysis of gendered practices in order to help explain the constructed social reality of the lab and its denizens.

From the previously established frameworks of materialism arises another key theoretical frame – that of Bourdieuan habitus (1977). By examining the history of women in computing, and by supplementing this history with ethnographic accounts of gendering within the lab, I argue that this habitus both exists and is maintained through indexical forms of discourse. Through this frame, women are implicitly excluded from the lab, either through being dismissed or ignored by their male peers.
The ENIAC Girls – Today and Now

According to historical accounts of computing, women were directly involved in the development of programming. Adele Goldstine, Betty Snyder Holberton, Fran Bilas, Kay McNulty, Ruth Lichterman, and Grace Hopper are all associated with different programs – some of which are either still present or precede those commonly used today. The anagrams form a veritable alphabet soup of accomplishments: ENIAC, COBOL, UNWAC, FORTRAN, etc (Light 2014: 72). In fact, what we today call computers were named for the person the machines came to replace – the “computer,” a person who computes. At the time of one machine computer’s birth, the majority of these human computers were women. These “ENIAC girls” (named for the project for which they were hired) were mostly recent college undergraduates with math degrees, hired during the Second World War to calculate ballistics and work on a project curated by John Mauchly and J. Presper Eckert (Light 2014: 64).

The Electronic Numerical Integrator and Computer (ENIAC) would be the world’s first “general purpose electronic” computer, specially programmed to calculate ballistics for the armed forces, yet also capable of handling a wide range of calculations beyond those required for weaponry (Goldstine 1946: 97). The war effort offered a moment when women could enter scientific fields previously inundated with men, but it was a controlled (and deliberately limited) entrance. New working roles, such as computing, were created for women in this period – but in such a way that female labor within the specialized fields did not pose a threat to male labor, which would return after the war concluded. Many of these jobs were specifically designated as ones for which women could apply – despite the ongoing male labor shortage, a gendered division of labor remained an aspect of the national workforce throughout the war (Light 2014: 63). This gendered division of labor allowed corporations and laboratories to render female
involvement “invisible” (Light 2014: 62), an invisibility that was further exacerbated by the credit for successful projects being publicly given to male engineers/inventors, instead of to the women who did most of the legwork (Light 2014: 73).

In the case of the ENIAC girls, their contributions as programmers soon rendered their new, “subprofessional” roles as computers obsolete; nonetheless, after the war, many of the women programmers were replaced with men, despite their experience, or forced to find new jobs as positions previously created by the war effort disappeared. These women, and most of the women with math degrees who followed them, were instead encouraged to become grade-school teachers (Light 2014: 72), while their objectified, “feminized clerical labor” (Light 2014: 60) was transformed and refined by the laborers who stayed behind, who were mostly male. These post-war male programmers came to inhabit and to shape the new workforce of computer scientists.

By contemporary accounts, the ENIAC program was a success – a success attributed to and developed by the Moore School of Electrical Engineering (Goldstine 1946: 97). The fact that the ENIAC girls, who operated within the Moore school, were never mentioned or publicized in print or photography ensured that, while it was their labor that made the program work, the “girls” never were recognized for their contributions to the social knowledge surrounding the development and implementations of this new technology (Light 2014: 68). Marx (1978: 285) argues that it is production itself which shapes society. This can be supplemented by considering the exploitation of laborers to be a process through which workers are alienated from their product and thereby individuated in favor of the product standing alone as objective value (Marx 1978: 262). By coupling these arguments, this moment of ENIAC’s birth was a moment that stripped the personhood from programming while obscuring – one could say mystifying – the
reality of the conditions of production. This means that the objectified product, ENIAC and those computers that followed it, could shape society through its existence and use, but because of the objectification and subsequent dismissal of early computing labor, the women who created the preconditions for the emerging technological revolution were forgotten. Female labor was what went in – and a functioning computing system was the result, supposedly achieved by the Moore School. Since acknowledgement was only given to the male engineers who designed the project, ENIAC was widely perceived to be a male success. Computer production was therefore perceived as wholly male. This could only be accomplished through the alienation of the feminine from the operative work that built the program, which required the objectification of this labor and exploitation of those women who worked to accomplish the dreams of the male engineers.

ENIAC was perceived to be the product of masculine labor through the invisibility (and objectification) of the female operators, yes, but there was a secondary moment in which this perception was reinforced. Material production is not all that produces – the consumption of the material good is another form of production (de Certeau 1984: xii). It was this secondary moment which helped establish computer science as something standing apart, quite separate from women’s work. What Light considers a contradiction between the work women did during the war and the way that work was subsequently perceived (2014: 67), is, directly, a conversion of female work into a neutral (non-historical, as Marx would say) object, one that can then be represented as the product only of male ideas through its social and physical consumption.

While the gendered social division of labor during the war was one that appeared to suggest that hardware should be coded male and software female (Light 2014: 67), software was later subsumed by the male-dominated industry and turned male through its representation to and
consumption by the public. After this suppression of the industry’s wartime history, all aspects of
the computer – software and hardware – were designated as male. To justify this new
representation of technology, the manipulation of computer hardware was attributed to a male
affinity for practical skills, while the manipulation of software was couched in terms that
represented male acuity as more suitable for computer work than female dexterity (Bray 2014:
374). Women were deemed inferior, both intellectually and physically. The production of this
logic was secondary to the development of the technology, but soon seen as inherent in the
consumption of the computer as product, after the physical production of computers began.
Women were given no accepted place within computer science in the immediate post-war period
– although vestiges of their initial labor remained within the very fabric of early computer
programs, vestiges of code that were invisible and inaccessible to all but a select few.

This social perception of the masculinity of computer work would come to shape the
historical importance of the ENIAC project. A precedent had been set that established a set of
gendered norms within the new technology. This is production on a grander scale than the
production of material objects. It is the production of “habitus” – that which is sensical in
society. History is habitus, as Bourdieu argues, which is converted to a semblance of nature,
becoming unconscious or inherent to those in a culture whose actions are shaped by it (Bourdieu
1977: 78). In this notion of habitus, Bourdieu refers to those operating – producing – within the
society in which the particular habitus exists. This means that historical information helps to
develop the deeper social structure through which actions and products are objectified, defined,
and thus made understandable and sensical within society (Bourdieu 1977: 79).

Since technology has had norms which could structure its habitus, it has been portrayed
as a field in which women have no place. It has become a matter of common sense, and therefore
expected, for the field to be most hospitable to men. The past is inherent in the present – the past shapes current social conditions (Bourdieu 1977: 78). Since this is the case, the consumption of history allows society to produce its own expectations for present society. As society perceives the field of technology – and later, the field of computer science – to be a province of the masculine, it reproduces the expectation of male dominance in the field by normalizing the occurrence of male domination within the field. This necessarily forgets the extensive contribution of women in the early moments of technology.

This brings us to the present. Women have infiltrated previously male dominated careers such as medicine and law (Faulkner 2000: 92). Yet within the field of computer science, a gendered divide still looms (Misa, 2010: 7), which suggests that the habitus introduced above is still ingrained in social understandings of technology and computer science. Women are underrepresented in computerized games (most of the characters are male), in the workforce, and especially in leadership positions. Of most concern is a “replicating pattern” of “fringe incidents” that target advocates for female and sexual equality – these targets, most of whom are female or do not otherwise fit into a heterosexual male identity, are viciously trolled and hacked, which some see as an attempt to intimidate and to normalize violence directed toward women and women’s contributions within the field (Hicks 2013: 88). Women working in large information technology firms such as Google are also coming forward with allegations of abuse and gender discrimination in the workplace (Wong 2018). If technology and its production is a point of power and leverage (Bray 2014: 372), then these “fringe incidents” show that technological power and leverage are still normalized as being held only by men. While this is currently being challenged in all fields through the #metoo movement, gender still has a role to play in the field of computer science. Through the targeted exclusion of women in the field, the ideal computer
scientist --at least to those in power in the industry—is reinforced both to the public and within the field as being a white (or perhaps Asian), heterosexual male.

To return to Sadie’s remark above, this gendering is seen everywhere in computer science, not just within the workforce. When Sadie describes having her own code explained to her, she is recounting a tangible example of female acuity being considered less sharp than male. While the incident could have been a simple misunderstanding, the fact that it stood out to Sadie indicates that she felt it to be a gendered instance of “mansplaining” (when a man explains a concept to a woman out of the condescending notion that she does not have prior knowledge of it) (Solnit 2014: 14). Sadie is currently a lab TA – a position that requires nomination by a professor and implies that she is earning grades comparable to, if not surpassing, those of her peers, the majority of whom are men. At the very least, this instance of “code-splaining” made her feel uncomfortable and frustrated, like she was not being heard or given the consideration of a male peer. If one’s perceived knowledge directly influences one’s social standing within the lab, this could have been perceived as a moment in which Sadie’s authority was being (however indirectly) denigrated. At such a moment, gender has a direct bearing on authority and the allocation of respect as was explored above – and that could have great potential for shaping the perception of women within both the individual lab and the integrated field.

The Master Theorem

These blatant moments in which women in computer science (and elsewhere in society) are not considered as competent as their male counterparts are similar to those in the case of the ENIAC girls: men are perceived to be the thinkers and inventors, and thus they must be the nodes from which authentic knowledge stems. The earlier historical accounts of ENIAC
dismissed the work done by women computers, while they also minimized the contributions made by the couple of female engineers on the team (Light 2014: 73). Although this dismissal originally began in the media coverage of ENIAC, the erasure also had its place in the lab. In cases such as this one, dismissal refers to not acknowledging the contribution of a person or persons in a collective endeavor. Since a person’s contribution in computer science is most often their knowledge made product (within code), a dismissal of programming expertise is tantamount to not acknowledging the knowledge of the dismissed, because it is assumed that such knowledge is lacking. As we now know, the ENIAC girls actually possessed an intimate knowledge of the machine that could not be translated to someone who designed the machine – for example, these early programmers were the first ones to know if a tube was malfunctioning, which tube it was, and how to fix it (Light 2014: 67). Their more embodied knowledge of processing allowed them to put the finishing touches on what was dreamed of by the original engineers.

Within the CS lab, there are no engineers versus computers. Interactions take place between and within student peer groups. Instead, knowledge in the lab is measured against peers’ self-reported programming successes. Moreover, this self-reporting is never pushed – men are not expected to show proof of their supposed competency. If anything, this makes the cases of dismissal both more ridiculous and more serious, since all students taking the same course (such as CS3) are considered to be at an even footing. In fact, every student interviewed acknowledged that it was in CS3 that any advantage gained through taking CS classes outside F&M disappears – it is uniformly difficult for every student. This means that when a dismissal occurs at this level, it is not coming from a place of previous experience but from one of preconceived notions.
I knew the four CS3 students at the front of the room were confused when the argument grew to involve Rahul. As if to settle whatever was causing the ruckus among them, Charles called out, “Yo, Rahul! What did you do for this assignment?”

Rahul looked up from where he sat in the front of the room at the lectern and lazily inquired about which assignment Charles was referring to. Charles offered a brief explanation, having been the one who initiated the tutor’s assistance. Everyone else kept his/her eyes on Rahul, including me. The tutor shrugged and told the group of students that he used “big theta,” along with a string of phrases that no one else seemed to understand.

“We’re using ‘big O,’” Charles interjected, more confused now than he was a few minutes earlier. “Where did you get theta?”

“Yeah – there is no theta in this equation,” Holly, the female student sitting in the front row, contributed.

Rahul, always confident, did not seem too perturbed at the confusion, simply answering that the theta comes from the Master Theorem, which obviously was the only feasible way to get through the assignment.

If anything, that made the confusion worse. None of the students had heard of the Master Theorem and were beginning to panic – the assignment was due in a few days. The professor had obviously not covered this in class. Rahul, rather unhelpfully, said the Master Theorem was a mathematical equation, but then the gravity of the situation seemed to register and he stood up to oversee what the students had been doing, if they were not using the correct equation. He looked over Holly’s shoulder for a minute as the three male students behind him began to type into Google. After about a minute, an exclamation from Holly drew everyone’s attention back to the
front. She was pointing at a PowerPoint slide – the Master Theorem had been in the lecture notes the entire time.

The reactions were immediate. “I don’t believe it,” Charles scoffed. He had not even looked up from his screen, even though Holly was sitting directly in front of him.

“He never covered that slide,” another student, sitting next to Charles, quickly defended, also looking at his screen.

Holly and Rahul, the ones facing me, looked suitably surprised at response. Holly even quickly looked over her shoulder to reaffirm that yes, she did read the slide correctly. She did not turn back around, obviously affected by the sharp dismissal. Instead, she began to talk quietly with Rahul, who was still standing next to her.

Meanwhile, the second guy seemed to find something. “Oh shit, thank you!” He said, still facing his screen. He had found a relevant website that addressed the equation – he had thanked his search luck—or, perhaps, he had thanked his own manipulation of the screen.

Despite his initial dismissal, Charles had pulled up the slides and had begun to move through them. I watched him slowly straighten in his seat before saying, “Wait, guys. This is actually relevant.” At this point, the two other guys now looked at the lecture slides being presented to them via Charles’s computer. Holly was not referenced, thanked (very notably unlike the screen had been), or included in the resultant conversation. In fact, when she turned around with a question after Rahul went to sit back at the lectern, one guy did not notice and instead turned to talk to the guy beside him while Charles heard her out and responded.

Lily then walked into the lab, wearing her coat and carrying her bag, most likely to greet the students in the space. Everyone noticed her entrance and began peppering her with questions about the Master Theorem, Rahul included, so she stopped for a moment to talk to Charles
before turning to address Holly, who was standing patiently to the side because she could not attract Lily’s attention from her chair.

This situation is relevant to our discussion mostly due to the speed and normalcy with which Holly, a CS3 student just like the other students in the interaction, was dismissed. Her ideas were heard, yet not believed by the men. There was a clear privileging of the ideas proposed by the men in the room: only when Charles deemed the slide relevant did it become so for the majority of those in the interaction. This is a gendered exclusion, made even more clearly so by the way Holly had to stand to the side and wait her turn to be noticed by the senior “expert” who entered the room. While the senior was also female, she did not challenge the status quo – she instead supported the patriarchal reality by first addressing the question posed by the loudest in the room, the male student.

At the most basic level, this exclusion is a rejection of Holly’s capacity for knowledge production in the CS Lab. The lack of attention to her discovery or peer acknowledgement of her initiative prevents her from directly contributing to the collective knowledge within the lab. Her contribution is instead attributed to another. Holly’s would-be contribution must, instead, be introduced by a male counterpart such as Charles to be accepted (albeit never acknowledged) by others. Not only does this prevent Holly from accumulating authority within the lab, but it also prevents Holly from accruing any publicly acknowledged competency from her male peers.

Jane Margolis and Allan Fischer hypothesize that this form of gender discrimination within Computer Science begins early and is rooted in a perceived difference of values (2002: 5). Over time – predating the birth of computers like ENIAC – technology has been strongly associated with masculinity. The logic behind this states the women are traditionally considered to be more interested in working with people, whereas men are more interested in working with
machines (Faulkner 2000: 93). Within this dichotomy is a further division, specifying which machines are acceptable for women to use – refrigerators and other household appliances fit into the category of “soft technology,” which is opposed to rockets, lasers, and computer code – “hard technology,” or “real” technology. Because “soft technology” is used in ubiquitous, domestic contexts, it is not even considered technology to those who use it. Moreover, since it is everywhere, it is not given the same value as “hard technology” does (Faulkner 2000: 93). When men are perceived to be more suited to technological innovation, then it stands to (gendered) reason that men are more suited to handle the power created by construction and use of that technology.

Women are therefore believed to excel in situations that call for “subjective rationality,” which requires a more holistic, emotionally-connected approach to problem solving, and which is useful when working with people. Directly opposed to this concept is “objectivist rationality,” which calls for emotional detachment, and an abstract or deductive form of problem solving. Men in western society are considered to be better attuned to this variety of rationality (Faulkner 2000: 94). The western dichotomization of gendered intellectual styles is deeply reinforced within the computer science field. In an interview for her new book, Brotopia, journalist Emily Chang alleges that the high percentage of men in Silicon Valley (which is 75% male) is due to a perception that links “good programming” to “not liking people” (quoted in Corbyn 2018).

Such gendered values are instilled in children at a young age, often through role-playing games that mimic the subject positions of adults: for example, play that establishes the young girls in the kitchen, while the young boys build things (Faulkner 2000: 94). This mimicry both establishes and reproduces a social norm that then structures what is commonplace and accepted
more widely in social life. Bourdieu’s definition of habitus hence offers an important perspective from which to view gendered interactions within the lab.

The Bourdieuan habitus structures a “commonsense world” with the “objectivity secured by consensus on the meaning (sens) of practices and the world” (Bourdieu 1977: 80). This encompassing habitus, determined by social conditions of the moment, has its “practical reality” in the early development of those members of society who it comes to shape (Bourdieu 1977: 78). Habitus exerts its maximum effect during childhood – precisely when the gendered disparity between technology creation and use becomes evident. The symbolic association between technology and masculinity is thus incorporated into what becomes sensical and regular within society at an early age. This suggests that situations as that described above do not become normalized in undergraduate computer science programs, but long before. However, the acceptance of what Erin initially identified as “microaggressions” reinforce the gendered habitus and hence the disparity in social production within the CS Lab space.

This reinforcement is a “coordination of practices” that only arises after the habitus has been well-established (Bourdieu 1977: 81). Such coordination occurs in an interaction – like the one above – between “social persons conjuncturally brought together,” who are themselves in “objective positions” within the established habitus (Bourdieu 1977: 82). Children, from an early age, are subjected to a habitus that normalizes male involvement with technology – specifically, with computers – while also rendering female exclusion as unthinkably acceptable and normal. The people who were once children engaged in habitus-reproducing activities then grow and learn to navigate social situations within the boundaries of the habitual framework – thus, social interactions arise, over time, from the habitus. Since male and female members of society are exposed to different, gendered pressures exerted by their habitus, they take on separate positions
that feel “objective” because of those practices’ deeply ingrained acceptance within society. A man is seen one way by society, notably as someone more capable of objective rationality and of emotional detachment (to return to Faulkner’s argument). A woman, on the other hand, is understood to be better suited for subjective rationality and emotional connectedness (Faulkner 2000: 94). Therefore, men are, *de facto*, more suited for technology work. This is objective in that it is not understood as circumstantial nor subjective within society – even if this is not true to lived reality, it has been established via historical precedence and considered natural, so it is perceived to be true.

Emotional detachment, in particular, has been identified as a characteristic of the “masculine ideal” as a social marker delineating male from female (Bird 1996: 125). Here, Bird refers to gender identity as a “continual process,” which relates to a Bourdieuian habitus in that this process repeatedly affirms the habitus, or norms, which then structure what is affirmed (Bird 1996: 122). This habitus situates itself around homosocial interactions, which become that which reaffirms the hegemonic masculine ideal, which is currently privileged in society (and in the field of Computer Science). Connell situates homosociality within a historical context – ergo, habitus (Connell & Messerschmidt 2005: 832). Now, the lab is not just a space in which women are excluded, but a place in which women cannot help but be excluded from – homosociality actively works to define and redefine competence to privilege the male and thereby “latently” functions to exclude women (Holgersson 2013: 61-4).

However, women still code. They still work for Google, Apple, Uber, and other tech companies. Methods of interaction are steeped in habitus, as Bourdieu posits. How are these women interacting and participating in the lab? At the beginning of this section, we see an example in which a woman is refused entry into the production of social knowledge within the
lab. This was a result of an unconscious divide that decouples femininity with technological competence and relates this to the specific “nature” of genders. This is habitus: history made nature.

However, for this divide to truly be successful, it must be continually affirmed. From habitus comes stereotype. Contributing to both, however, is the constant ebb and flow of interactions within the lab – discourse.

On Discourse

The three men at the front of the room were discussing gaming. It was the first mention I had heard of gaming from the group, so I was trying to pay close attention from where I sat, two rows behind them. We four were the only ones in the lab at the time – I was the only woman.

“You play lately?”

“Nah, man,” Eric answered, not looking up from his screen. “I couldn’t get into it.”

“What?” Charles seemed incredulous, turning to give his friend a look.

“You can’t customize characters,” Eric continued, still typing. “I don’t like games where I can’t customize the characters. And I don’t like the guy characters they have.”

“You could be a lady dude, then,” Charles quickly replies, and they all laugh. The thought of playing a female character (albeit not fully female – a male playing female, therefore “lady dude”) seemed to be a ridiculous suggestion. There also seemed to be little to no consideration of the fact that Eric is one of the few black students in the Department – the stock character in question might not have been true to his racial identity.

Charles then darts a glance over his shoulder, almost reflexively, and briefly catches my eye. The laughter dies off awkwardly, and everyone returns to work.
My invisibility within the lab offered certain perks, as seen above. My position as an anthropology student made me feel fortunate that the CS students spoke openly with each other in my presence. On the other hand, my positionality as a female student within this context could not have been clearer, compounded with my realization that this positionality mirrored the positionality of any other female CS student using the space. In such an intimate learning space as a classroom (intimate in its close quarters and its design as a space for cohesion), some students are nonetheless simply not seen. These students are mostly female. Some of these students outperform their male peers in the CS field. Others become literally invisible, either not using the lab (and therefore not contributing to the social knowledge being generated within the lab) or dropping out of the department after taking CS1. No one I talked to was able to pinpoint a single reason for this drop-out phenomenon, but some upperclassmen had theories.

The disappearance of women such as these beginning CS students—part of the classic “leaky pipeline” (Blickenstaff 2005)—is a national problem. When I asked him about it, Rahul, the CS3 tutor, responded with a shrug. As a whole, he said, the ratio at Franklin & Marshall was “getting better” (becoming more equitable). His rationale was that the women who start at CS1 with the intent of majoring are the ones who stick it out—the others either become more enamored with another subject or were only taking the course for general education/interest purposes. The women I talked to seem to agree. For instance, Meredith, a CS1 student, knew she wanted to major in CS after taking classes in high school. Meanwhile, one of her peers said that CS1 was much harder than she had anticipated, and that she would instead be focusing her studies within the Film and Media Studies Department.

While at first glance Rahul’s argument seems valid enough, there are issues that necessarily need to be addressed before one can fully accept it. Rahul himself came to F&M
knowing he would major in CS after taking computing classes in high school, but he will be graduating with a double major in CS and Business (BOS). Billy, the senior crafting the network which claims to understand the emotional valence of tweets, admitted that he was initially planning on majoring in Economics with a minor in CS. He ended up having a strong dislike for the first class he took in the Economics Department. Now, he’s strictly majoring in Computer Science. He did comment on Rahul’s presence in the lab, saying that he had doubts as to whether or not Rahul was as committed to CS as he was to business – it seems to be common knowledge that Rahul wants to become a businessman after he graduates in May.

While David was in slight agreement with Rahul by implying that only CS majors are justified in their involvement within the lab space, he himself did not enter the Department with intent to major. Dave, a junior major, did not even know what CS would entail at the beginning of his college career– he took the class as a filler class because he liked to tinker with the mechanics of technology. Similarly, Lily, one of the senior students, originally intended to major in Math, and only happened upon CS1 when she needed an extra class to fill her schedule. Why, then, is it justifiable to state that “only women who enter the Department intending to major stick it out” while anomalies exist on both sides?

When asked what the appeal for Computer Science was (as opposed to the other majors offered on campus), the older students all said they appreciated how close the Department was. Billy argued that instead of CS being a competitive space, there was teamwork and a level of sociality which appealed to him. Rahul, although already having known that he would major in the subject, separately reaffirmed Billy’s statement by commenting that “[getting through CS3] is so hard, it’s basically impossible to do it alone.” This led him to appreciate the existence of a socially collaborative space such as the Lab, even though he alleged that he worked
independently from the lab until CS3. Dave found the lab space to be comforting and would even go there to work on Friday nights, when it was empty, implying that he too enjoyed the space.

This praise for the social quality of the CS Lab was not a sentiment shared by the (mostly younger) female students. Meredith confided that she would go to the lab primarily to meet up with Sadie, who would help her with assignments. Meredith and Sadie are in the same sorority, hence their familiarity with each other. Their network external to the lab has given them the comfort with each other that can then be transplanted into the lab – while they may have become closer on account of their shared academic interest, it was not the lab that initiated that close bond. For women to have bonds with one another, then, requires some other social network on campus. Notably, the social network in this case is an all-female organization – a space in which women are actively encouraged to inculcate bonds with their “sisters.”

As mentioned before, most of the CS1 girls use the space for access to the tutor. This implies that the space does not have the same level of comfort or sociality for women as that which appeals so strongly to the male students. Owing to the lack of comfort, perhaps, is the lack of recognition women garner in the lab. Harkening back to the discussion on greeting styles that opened this chapter, we see that women tend to slip in and out of the space undetected by all in it, their close friends (most of whom are also female) excepting. Even once they are in the space, they also have to work harder than their male counterparts to be recognized by the tutors and their male peers. Even if the room is completely full, a comment or question made by a woman can go unnoticed – or ignored. While no women with whom I talked personally mentioned this point (suggesting that it has become so expected that it might not bear mention), their lack of attention in the lab struck me as blatantly off, especially in a place that the male students heralded as being a refuge in which to develop relationships with other students.
I recorded this particular phenomenon in my field notes on a night such as that described above – the lab was packed, and people – mostly male – were filling up every row. Sadie walked in, seemed surprised to see everyone clustered around, and made a comment: “Wow, it’s a party in here!” No one looked up or responded verbally, but since this was rhetorical to begin with, she did not seem bothered by the lack of attention. Instead, she weaved through the rows and sat right in front of me, in the middle of the third row, next to the CS2 tutor – the most central spot in the lab. She reached into her bag and pulled out her laptop, notebook, and phone.

At that point, she paused. “Does anyone have an iPhone charger I could borrow?” she asked, directing the question to the group sitting near her, all men in CS2/CS3. (In other words, all men in her class.) No one responded, even as she waited.

I cleared my throat and dug mine out of my bag, saying “here” and handing it to her. She smiled and thanked me in her friendly manner.

Yes, it was loud, and perhaps no one (besides me) heard her. However, I was also paying attention to her. Charles had looked around, briefly, as she took a seat next to him, but was clearly more absorbed in the questions being asked by the male CS2 students to his right. At that moment, Sadie did not warrant his attention.

When contrasted with male behavior in the space – calling out to older students, as seen with greetings, beckoning when they need help from the tutor, joking and carrying on conversations while working – the women fade discreetly into the background. They raise their hands and patiently wait for the tutor to recognize that they need help. They do not call out to others. Instead they hold muffled conversations about their work, how they feel about the semester, etc. They do not always sit by gender – it is perfectly common to see a male and
female student working together. This being said, it is rare to catch more than two women using
the space at any given moment, unless it is a Monday night and the CS1 tutor is present.

What seems to happen in spaces dominated by men is the creation of a discursive mode
that becomes distinctly male-centered, which becomes the norm when it is reinforced as habitus
within the lab. The masculine discourse type is then authorized through its use by all in the space
and accepted as a “natural” part of the lab, simply as what makes a successful computer scientist.
Such masculine modes of discourse therefore express the true habitus that regulates the space,
usually to the detriment of the experiences of the people using the space who do not hold the
gender-privileged position. The fact that one sort of discourse is consistently recognized and
normalized, and the fact that that discourse is what is being exchanged almost exclusively
between the men in the lab, sanctions this behavior, along with the space in which this behavior
occurs, as male. Therefore, a masculine gender discursive code must be adhered to, in order to
keep normalized “order” within the space (McElhinny 2005: 25).

How is this masculine/CS gender order being defined? In the case of discourse, gender
“barriers” are constructed through the social and discursive interactions between different
groups. Not only do moments in which female speech acts are ignored or not heard abnegate the
presence and the place of women in the lab, but such invisibility consequently reaffirms the
speech and other acts of men. Women’s involuntary silencing then makes the men sound louder
– and thus more important. Ridgeway explores this particular phenomenon as she discusses the
broader category of “status beliefs,” which are based on historical perceptions that situate one
group as necessarily superior to another (2002: 637). Without a doubt, discursive practices are
inculcated into gendered status beliefs – certain modes of discourse, learned early through
socialization (and habituation), are distinctly “female,” as opposed to “male.”
This categorical distinction has been closely examined in feminist discourse theory. Language and gender, according to McElhinney, have an “indexical” relationship. Modes of language become attributed to gender through the form of habituation referenced above. If a person practices or says something that is considered “female,” the person is then considered to be female. The use of language, in this context, points to the gender of the speaker (McElhinney 2005: 35). Gender is construed by the listener, though – [s]he assigns “situational meaning” to the words, which then create an understanding of the gender of the speaker (McElhinney 2005: 35). For this reason, there can be multiple interpretations of one utterance. Too much variance in interpretation causes confusion. However, since the interpretations are made by those already habituated to the society that informs these indexes, there is a general uniformity of understanding. Through this logic, language is that which constructs gender, not the other way around, countering the notion that gender is something one can “have” (Kelan 2010: 177; McElhinney 2005: 24).

Here, there are echoes of Foucault – diffuse power regulates and sanctions a “discursive explosion” around a given topic (for Foucault, this is sex) (Foucault 1978: 18). This then required sex to be spoken, to be continually monitored and policed, and, as a result, perversions previously undefined were concretized, forced into language, to create a distinction between “licit” and “illicit” sexualities (Foucault 1978: 37). Homosexuality was then concretized, compared to its analog, heterosexuality, and considered inherent to the person displaying that sexuality. Traits such as sexuality, then, are not merely inherent but maintained through discursive distinctions.

While his was not a theory of gender, the same basic principle can be related to the gendered discourse theory outlined above. Discourse is a powerful tool that can be used to create
differences through its utility. Indexicality is the continual process by which these gender differences are maintained. Having cues that one associates as being “male” or “female” enables one to make sense of the gender of those around. This is especially salient in the lab. Different patterns of speech (or silence) emerge among the students using the space – these differences closely align along gendered lines. By posing a question – “Hey, does anyone have a charger?” – and then waiting, Sadie indexed her gender to be not male, and therefore female. The male approach to everyone ignoring his question would be to ask it again, directly, calling the name of a specific individual or group to catch the attention of those around him.

The discursive practices of women in the lab generally involved silence, as seen above. I never witnessed a female student call out to a tutor for help – instead, they still insisted on raising their hands or similarly signaling to the tutor, who would then come to them. Even resistance was gestural – remember Sadie’s comment at the beginning of this chapter, in which she threatened to leave the lab after facing a discriminatory remark (as opposed to asserting herself in an argument). These forms of gestural indexes were not seen among their male counterparts, especially the upperclassmen. Instead, these students would be boisterous and call upon one another for help. They were not afraid to raise their voices (and gain the attention of all around them in the process). Female silence has been noted and attributed to Western cultural norms that expect women to be silent, especially when compared to men (Steinem 1998: 336).

The women in the lab are then participating in discursive modes that they have been habituated to outside the lab, when they were younger. These is classic female discourse which is trying to find ground in a space characterized by classic male discourse.

In addition to the raised voices, men in the lab were often verbally competitive, as realized in the previous chapter. While vocal interactions in which they compared others’ work
to their own helped establish a hierarchy among the men, it also created a pattern that newcomers were expected to fall into, should they wish to be accepted into the space. Competition is a prevalent theme in male-male interactions, as opposed to cooperation, which is typically gendered as female (Cameron 1998: 208; Bird 1996: 122). To not partake in competitive talk would then be to not fit into the masculine space. By default, this indexes one as “not-male,” and therefore places one outside of the masculine discursive practices that are privileged in the lab and crucial in the hierarchization that happens in the space.

These differences in discourses create a “gendered spectrum of belonging,” a term I borrow from Alfrey and Twine (2017: 31). Since most of those who use the space (men) communicate using the same discursive style, this style is then allocated privilege within the space – it becomes the primary method by which ideas are conveyed. This plays into the “system of prestige” on which CS operates. Since the technical skills required to code are self-taught or learned on the job, some other evaluation is necessary when distinguishing between those who belong among the ranks of coders and those who do not (Alfrey and Twine 2017: 31).

Unfortunately for the women, this distinction within CS targets discourse patterns that they might be comfortable with, such as cooperation.

The distinction that arises between those who partake in dominant discourse and those who do not renders gender as relational – women seem more female when they do not take part in the masculine discourses (Bird 1996: 122). Breaking the “rules” of male discourse, Fine argues, is to “underline the existence of these rules and [rule-breaker’s] incompetence (culturally)” (1987: 145). Competency is then configured along the lines of how that competency is discursively communicated. Trying to argue one’s competency in discursive patterns indexed
as “female” is futile – not only does it fail to communicate the point because competency is not understood in those terms, but it also serves to reinforce the pre-existing discursive pattern.

Perhaps for this reason, gender-fluid women are more able to assimilate to male teams and maintain their status (and accompanying competency) while doing so. Since they do not consider themselves fully female, they have fewer discursive barriers to overcome to fully participate on a male-dominated field (Alfrey and Twine 2017: 31). This itself can be problematic, though – the notion that discourse will only index along a gender binary places all who do not personally identify with either gender in an interstitial place that cannot be made sensible through the “hegemonic discourse” (Kelan 2010: 180). Even women who consider themselves female will develop strategies to be seen as less feminine, such as by downplaying their femininity through their dress or discursive positioning (Alfrey and Twine 2017: 30). However, taking part in these discourses could also distance women from their female counterparts while creating complicity which reinforces the normalized masculine framework. To be “one of the boys” requires that a woman acts and speaks in a way that minimizes the disruption her unfamiliar discursive practice imposes on the male discourse that rules the space (Fine 1987: 145). To be acknowledged as someone with authority in the space is to adapt to the previously established norms and work within those. Many ethnographies in male-dominated fields – such as business and restaurant kitchens – identify the exclusion of women as being due to discursive practices which do not align with the norm (Holgersson 2013; Fine 1987).

Unfortunately, even adapting and adopting male discursive patterns does not guarantee that a woman will actually become “one of the boys.” Competitive talk and joking among men are ways in which men communicate their “masculinity” in a direct effort to distance themselves from being “effeminate.” However, as women try to adopt these mannerisms to relate to their
male counterparts, they are not becoming more masculine as a result – instead, they are just becoming “unfeminine” (Miller, Casey, and Swift 2000: 292). No matter how hard they might try, women will never fully become masculine – therefore, they will never truly be accepted into the masculine discourse, although they may be hailed an “exception” for making the effort to fit in with the popular male discourse (Fine 1987: 144). They can also be exceptional if they display that their competency far out-paces that of the men on their team (Ridgeway 2002: 646-7). In other words, if their genius cannot be denied, it is acknowledged. In both these instances, women transcend the previously established gender distinction – they become not male, yet not truly female. Respected, yet not truly accepted.

Within the lab, Lily is this exception. She does not take part in masculine discursive practices – in fact, I rarely saw her in Stager 002, except on rare occasions when she would stop in to say hello, or if we had already made plans to meet (i.e. for an interview). When students had questions (as Holly and Charles did during the Master Theorem anecdote in the previous chapter), she listened to them and did her best to address the concerns. When contrasting this to Jack, who breezily managed to escape answering the questions posed to him through his assertion that he managed to “figure it out” on his own, it becomes clear that there are different discursive patterns at play. Lily used the opportunity to collaborate and ensure there was understanding of an issue – Jack levied a challenge, implying that to truly succeed in the Department was by proving oneself (i.e. by competing with others to attain the goal). And, because this is how success has come to be understood within the lab, the young men earnestly resolved to also prove themselves, to one day become like Jack – all the while not noticing the spaces made emptier by the lack of women who wish to subject themselves to an unfamiliar, exclusionary mode of discourse.
Conclusion

Space, epistemology, and gender all to inform the experiences of those majoring in Computer Science at Franklin & Marshall College. As students are disciplined to use the space in certain, prescribed ways through exercises and supervision by the tutors, they learn how to properly produce within both the physical and digital spaces. These spaces then dictate the forms of production and thus the ideologies that come to fill them, influencing social interactions between the students. From space, then, comes habitus – and given the history of computer science, this can be extended to argue that from the utility of the space comes the gender distinctions which actively, yet implicitly, work to exclude women from this space. Perhaps for this reason, there is such a low retention rate of female students within the Computer Science Department.

Also at work, and also mediated by production, is a process of hierarchization which is heavily dependent on notions of authority. These, in turn, stem from a student’s perceived competency within the lab. This hierarchization is mediated through male-centered discursive practices which prize classic Western male values such as competition and aggression – female discursive practices, on the other hand, have no place within this dynamic and are thus simply ignored, placing women outside evaluations of competency and thus neglecting to grant their work any authority when it is compared to that of their male counterparts. Discursive practices are indexed and are thus interrelated with gender in a way that types the speaker as either “masculine” or “not-masculine” (and therefore feminine).

While it is disheartening to witness the exclusion of women from such a rapidly advancing (and highly prestigious) field, this project is especially salient for an age in which women are beginning to demand equality, both in the workforce and in their education. How we
discipline and habituate students has deep implications for the gender composition of the workforce as well as for the well-being of those women working in male-dominated fields – within all the articles and ethnographic episodes cited in this thesis, there was nothing to suggest that women enjoy or are comfortable with being ignored or dismissed by their male peers.

A continuation of this project would evaluate collaboration and the role of explicit authority figures within the lab – while students do sometimes collaborate on projects (an activity that is generally typed as being “female”), this was difficult to examine while conducting fieldwork in the space due to its infrequency. Additionally, I frequently mention Professors throughout the piece, but due to being refused interviews with those at Franklin & Marshall College, I was unable to incorporate more than what I learned second-hand through the comments made by the students I interviewed. Understanding the agency an individual Professor might have over this space (and its subsequent social nature) could reveal additional areas of consideration for each of the three themes I outline in the body of this work.

It is difficult to predict how the #metoo movement will affect gender differences in the workforce or on campus. Similarly, it is difficult to predict the women’s place within the field of information technology and coding – perhaps, as home computers become more widely used and technology becomes destigmatized through its ubiquity, more women will be socialized to view coding as a skill that can be held by both men and women equally. Perhaps by then, AI has taken over and coding, as a mode of production, is rendered obsolete, completely nullifying these concerns of gender within the workforce. Until then, those women in male-dominated fields can expect to be subject to the same pattern of exclusion via the privileging of masculine discursive forms as was seen in Stager 002 – not from any animosity on the part of the male students, but
simply due to a process of habituation which normalizes these occurrences until no one is left to notice the missing women in the lab.
Bibliography

Abbate, Janet.

Alfrey, Lauren and France Winddance Twine.

Benoit-Barné, Chantal and François Cooren

Bird, Sharon R

Blickenstaff, Jacob
2005 “Women and Science Careers: Leaky Pipeline or Gender Filter?” *Gender and Education*, 17(4): 369-386.

Bourdieu, Pierre

Bray, Francesca

Brzezinski, Mika
2017 “With #MeToo, we need a serious talk about workplace ethics.” *Know Your Value*. Dec. 18.

Cameron, Deborah

Certeau, Michel de

ComputerScience.org

2018 “Women in Computer Science: Getting Involved in STEM.”
https://www.computerscience.org/resources/women-in-computer-science/

Connell, R.W and James Messerschmidt


Corbyn, Zoe


Faulkner, Wendy


Fine, Gary Alan


Foucault, Michel


Frazer, Sarah


Goldstine, H.H. and Adele Goldstine


Hall, Edward T.


Hammeren, Nils & Thomas Johansson

Hayes, Caroline Clarke

Hicks, Marie

Holgersson, Charlotte

Horst, Heather A. and Daniel Miller

Jasanoff, Sheila

Karanović, Jelena

Lave, Jean & Etienne Wenger

Lefebvre, Henri

Light, Jennifer
Lyman, Peter

Marcuse, Herbert

Margolis, Jane, and Allan Fischer

Marx, Karl

Massey, Doreen
1994 Space, Place, and Gender. University of Minnesota Press

McElhinney, Bonnie

Mellström, Ulf

Miller, Casey and Kate Swift

Michigan Tech
Niemi, Nancy S.


Oost, Ellen van


Peiris, D. Remanee, Peter Gregor, and V. Indigo


Ridgeway, Cecilia


Soja, Edward W.


Solnit, Rebecca


Steinem, Gloria


Vu, Shana


Wong, Julia Carrie