Chapter 3: Coding Black Functions for White Software Programs

> Google posts its job announcements on its website, so in theory, anyone with access to the Internet has the opportunity to apply. In practice, though, the jobs are closed to all but a small minority of people who have the education, experience, and personal contacts to pass extensive rounds of interviews and aptitude tests. I know many low-income people who would like nothing more than a well-paying job at a global technology company. But it doesn't matter whether they can browse engineering jobs on their phones. Online opportunity isn't always actual opportunity.
>
> – Kentaro Toyama, *Geek Heresy: Rescuing Social Change from the Cult of Technology*

Seventeen-year-old Trayvon Martin traveled with his father to Sanford, Florida to visit his future stepmother. That night, Martin walked to the convenience store to buy Skittles and watermelon juice. Neighborhood watch guard George Zimmerman spotted Martin walking back to the townhouse from the store and called 911. He claimed this hoodie-wearing man was on drugs. Zimmerman disregarded the dispatchers' warnings to not pursue Martin; he chased after the teenager and, after a struggled, gunned him down. As evidence, the hoodie did little to put Zimmerman in prison: he was acquitted in 2013, and in 2015 the Department of Justice decided to not charge him for violating Martin's civil rights. The federal government returned the hoodie to Trayvon's father in Florida with the Skittles and the juice. Finally, on August 21, 2021, the National Museum of African American History and Culture placed Trayvon's possessions on display (Roig-Franzia, 2022). The hoodie had more success as a symbol of civil rights activism against extrajudicial execution of unarmed Black men and boys. Trayvon Martin's death animated Van Jones, President Barack Obama's advisor, and the legendary musician Prince to found what was then called #YesWeCode, an initiative to train 100,000 Black youth in computer programming. Martin's hoodie became the central image of their efforts. Van Jones launched the project at the 20th Anniversary Essence Festival in 2014, where he shared the following story of #YesWeCode's birth:

> After the Trayvon Martin verdict I was talking to Prince and he said, "You know, every time people see a young black man wearing a hoodie, they think, he's a thug. But if they see a young white guy wearing a hoodie they think, oh that might be Mark Zuckerberg. That might be a dot-com billionaire." I said, "Well, yeah, Prince that's true but that's because of racism." And he said, 'No, it's because we have not produced enough black Mark Zuckerbergs. That's on us. That's on us. To deal with what we're not doing to get our young people prepared to be a part of this new information economy." (Rebuild The Dream, 2016)

Jones and Prince did make a good point: the opportunity to learn computer programming is an opportunity of profound transformation. Gas station attendants and retail associates can transform into one of the most highly valued positions in our digital economy. That isn't a metaphor but a real possibility, as most Black people in the private sector are frontline workers in three industries that offer less pay and fewer opportunities for upward social mobility: healthcare, retail, and accommodation and food service (Mckinsey & Company, 2021). Although computer programming is an emerging, specialized type of writing, public discourse suggests it's not really complicated if the factory worker or coal miner with basic computer knowledge can pick it up. The ease of access to and learning of coding literacy reinforces the idea that coding can be a driver for social mobility, which is endemic in conversations about democratizing computer science education. This *dream* of transformation is the heart of pipeline rhetoric.

However, when the appeal of "transforming" low-income workers into software developers applies to racially marginalized people, the imagined narrative above brings to bear legacies of racism and education. Historically, education for Black youth and adults in the United States has included implicit and explicit efforts to assimilate them into white middle-class society, to erase Black language practices and cultures, and to teach anti-Black linguistic racism (Baker-Bell, 2020; Gere et al., 2021). Public schools and universities continue this legacy of undue violence against Black people (Green et al., 2018; Hardaway et al., 2019; Joseph-Salisbury, 2019; Solorzano et al., 2000; Yosso et al., 2009). The student loan debt accumulated for attending universities adds to, not relieves, the financial strains of historically excluded people (Seamster & Charron-Chénier, 2017). And even after getting through college, the workplace welcomes Black people into more racial violence (Bohonos, 2021). Black people in the United States can live their entire lives moving from one anti-Black space to another with no chance of ever achieving the liberation

and well-being so long desired. Educational institutions cannot hide behind claims of distributing literacy for social mobility while supporting anti-Black policies, practices, and outcomes.

Chapter 3 arrests attention on computer code bootcamps as educational institutions that must also grapple with the reality that race and racism determine how Black adult learners and their instructors do and do not leverage coding literacy into social mobility. In this chapter, I argue that computer code bootcamps are racial organizations whose curricula and assessment practices "program" racially marginalized people into viable bits of code called *functions* to assist in designing white software systems—the technologies that largely center white end users and uphold white supremacist policies and practices. I analyze focus group interviews with Clearwater Academy instructors Richard and Jessica and Black adult learners and my own participant observation to understand how the computer code bootcamps' curriculum and assessment practices rhetorically shape their lived experiences. Clearwater Academy's curriculum and assessment suggests that Richard and Jessica must balance the needs of tech employers with the mission of ending racism and poverty. However, living according to these designs led Black adult learners to question the intentions and approaches of career training programs; they proposed an alternative curriculum design based on their own knowledge and lived experiences to create a Black coding literacy.

An empirical look at how racially marginalized people experience curriculum and assessment helps identify the limitations of those designs and the consequences those limitations can have on adult learners and even the instructors. The coding movement must contend with these investigations to achieve its proposed outcome of including marginalized populations in a profession that has predominantly been the space built for white men. In a study on the perspectives of Black Indigenous People of Color (BIPOC) computer science teachers, researchers Ivey et. al. (2021) argue that despite multiple conclusions that computer science education needs to teach critical digital literacies, the movement defines inclusion as access. Stopping at giving access to the tools of computer programming still "predominantly centers a Eurocentric perspective with little attention paid to the teachers (or students) who exist outside of the mythical computing identity norm of white, middle class, and male and 'color-blind rhetoric'" (n.p.). Not paying attention to these marginalized instructors and adult learners makes computer science, and training schools like computer code bootcamps, inadequate educational institutions for their needs.

In this chapter, I introduce racial organizations as a theoretical concept to better understand the pipeline rhetoric of coding movements as an interlocking *system of intra-organizations* that use coding literacy to help strengthen racial order in the United States. I use "coding function" as a metaphor

for writing (Alexander et al., 2020) to help literacy scholars study computer programming as a racialized literacy that *acts* on our bodies and minds when we engage with it through curricula design and assessment practices. Computerscience education continues to expand in public schools while enrollment among diverse students increases in university computer science programs (National Academies of Sciences, Engineering, and Medicine, 2018; CSforALL, 2021). Because the coding movement has expanded in the years since my study concluded in 2018, racial organizations and coding functions remain relevant concepts. Both may help computer code bootcamps and other computer science and literacy educators probe their intentions for coding literacy curricula, their approach to teaching coding, and what about coding we are teaching to racially marginalized learners, youth or adults. Unlocking interested parties' influence on computer code bootcamps especially helps them interrogate the ways we have brought the legacy of racializing written and oral language practices into computer programming.

Racial Organizations and Literacy as White Property and Economic Resource

Literacy scholars understand that reading and writing are a set of actions or behaviors; they are also a material resource tied to understood sociocultural meanings. In other words, we interact with literacy through objects like immigration papers (Vieira, 2016), education diplomas, and laptops (Vieira, 2019) and we give them significant meaning to navigate various social contexts (Brandt & Clinton, 2002; Burnett et al., 2014). The materials of literacy also have money-making power: In *Literacy in American Lives* literacy researcher Deborah Brandt (2001) observes that

> The nature of work in the United States puts a premium on the ability to traffic in symbols generally and in verbal symbols particularly, as print and print-based technologies have penetrated into virtually all aspects of money making. In an information economy, reading and writing serve as input, output, and conduit for producing profit and winning economic advantage (p. 25)

Brandt's observation that performing literacy practices with symbol systems that turn knowledge into profit contextualizes literacy sponsors (people but also institutions like computer code bootcamps) as managing the resources of literacy. They determine who accesses those resources and what rewards literate people accrue from leveraging those resources. Social, cultural, and economic demands can shift the materials of literacy and its value, and "As

literacy standards change, some people are economically lifted (think computer coders), and others are left behind (think typists). Keeping up with changing literacy standards requires investment—investment that depending on age, gender, race, social class, and other positions—is not equally accessible" (Vieira et al., 2020, p. 47). Opportunities to learn literacy, "in school and out," writes Brandt, "takes place within systems of unequal subsidy and unequal reward—systems that range beyond the influence of any individual family's assets, beyond any one pile of cultural capital that a adult learner or a home might accumulate" (2001, p. 170).

Studies on literacy can show how institutional racism provides a framework for governing the output and reward of literacy as an economic and material resource, often establishing white people as the most deserving of literacy's power and privilege. For example, literacy researcher Catherine Prendergast (2002) analyzes three significant Supreme Court cases on educational policy to demonstrate that literacy belongs to white racial identities in the United States: *Brown v. Board of Education* (1954), *Washington v. Davis* (1976), and *Regents of the University of California v. Bakke* (1978). While *Brown* eliminated racial segregation based in part on the psychological harm it causes Black children, the Supreme Court argued that Black children would fare better by attending integrated schools. This decision upheld the perspective that whites were superior to Black people and, more important, the superiority of the literacies they learn and use in public schools. In *Washington v. Davis*, the Supreme Court backtracked on the value of literacy. The justices questioned whether a high school diploma alone was sufficient for professional employability since the standards of education, they believed, had declined in recent years. The justices ruled that an exam that disproportionately disqualified Black applicants from police officer positions was fair because the exam was a necessary response to so-called poor education. Finally, during the *Regents of the University of California v. Bakke* both the defense for UC Davis Medical School and some of the Supreme Court justices realized they were alumni of Harvard University. Both shared an interest in protecting and upholding the institution's standards of literacy; they also pushed Harvard's own admissions program forward as a fair example and "determiner of constitutionality" (Prendergast, 2002, p. 225). Like *Washington*, *Bakke* questioned the need for affirmative action if more Black people entered these top schools and did well in their academic studies; hence, consideration of race in a university known for racial discrimination seemed unnecessary. Prendergast concludes that "Once remedy is granted in one literacy environment, that literacy environment is denigrated to devalue its worth. This is the economy of literacy as white property, an economy that served the white majority in the Supreme Court in its efforts to bring the course of racial justice to a halt" (Prendergast, 2002, p. 227).

Literacy scholars are well-positioned to study institutional racism, or how systemic policies and laws provide different kinds of access to literacy based on various social markers, especially race and ethnicity. However, sometimes literacy scholars—myself included—focus on how a single organization represents the macro-level of institutional racism. There's an opportunity to complicate our research by studying how systems of intra-organizations use literacy to construct race in the United States. With this approach, we widen our critical inquiry to how literacy sponsors are *collectively linked* in determining the life outcomes of their students, trainees, and workers and how these subjects collectively change those same institutions (or not). The coding movement's pipeline rhetoric provides an opportunity to learn the relationship between computer code bootcamps and the tech industry under the lens of critical race and digital studies. This approach helps me learn how computer code bootcamps participate in constructions of race in the United States.

Racial organizational theory provides an analytical rubric for doing this analysis. Sociologist Victor Ray (2019) argues that scholars of race and ethnicity have opportunities to uncover how racial ideologies operate through material resources. He observes that when race and ethnicity scholars study organizations, they can mistake institutional racism and organizational racism as one and the same (i.e. laws, policies, and practices partnering with school systems). These two belong to a three-tiered system of bigotry, in which school systems and workplaces are positioned between individual biases and the laws, policies, and practices that structure societal racial order. What goes under-examined is how organizations in this middle, or meso-level, tier contribute to the mundane reproduction of racial stratification. According to Ray (2019), "Individual racial attitudes and discrimination are enabled or constrained by organizational routines. More than a mere 'link' between macro- and micro-level processes, organizations are key to stability and change for the entire racial order. Organizations magnify the power and depth of racial projects and are a primary terrain of racial contestation" (p. 30). In other words, rather than contextualizing higher education, public schools, and career training academies as working within institutional racism, race and ethnic studies scholars should study these organizations as standalone sites that influence individual racist behavior and societal racial order.

Racial organizations amplify schemas about how material and social resources should be distributed among racial groups, and they amplify ideologies that justify this racial structure. Organizations consolidate power and resources and build them into social interactions that replicate these established racial structures and beliefs and influence the racial (institutional) order of society. In addition, racial organizations can shape "motivation, agency, and action in relation to resources and schema" (Ray, 2019, p. 35) in three

ways: First, they can unequally distribute resources, valuing white organizations over people of color organizations (such as the significant gap in grant funding between Historically Black Colleges and Universities [HBCUs] and Predominantly White Institutions [PWIs}); second, they require whiteness as a credential to access organizational resources. For example, a 2014 study on the relationship among race, gender, and criminal background in hiring for entry-level jobs found that hiring managers were more likely to interview white people with a criminal record than Black people without a criminal record (Decker et al., 2014); and third, racial organizations separate commitments to diversity, equity, and inclusion (DEI) from the practices and policies that lead to DEI. Thus, "objective rules and practices may be enforced in ways that disadvantage non-Whites[sic], or rules aimed at diversifying or ending discrimination may be ignored. This decoupling allows organizations to maintain legitimacy and appear neutral or even progressive while doing little to intervene in pervasive patterns of racial inequality" (Ray, 2019, p. 42).

Racial organization theory shows how the racial state influences the practices and policies of meso-level organizations and vice versa; the two aren't the same but rather are partners in determining how racially marginalized people navigate their lived experiences within and outside of work. More than racial ideology, social and material resources owned by such organizations shape people's working and private lives, with an interest in preserving and legitimizing whiteness. To this end, racial organizations can also re-direct racially marginalized people's agency for a specific purpose, one that sends them to the bottom of organizational hierarchy. However, subjects of racial organizations are not helpless or passive. Just as racial organizations can support existing societal racial order, they can also change the racial order for the better, sometimes thanks to on-the-groundwork of racial groups. Social movements, changes in policy, and reliance on the state "result from altering schema-resource couplings" (Ray, 2019, p. 43). Revising racial associations with resources within organizations may revolutionize the societal racial order. This not only happens in organizations; organizations can interfere with the policies and practices of one another, as well. This last point is a key focus for my analysis of the relationship between computer code bootcamps and tech companies later in this chapter.

The introduction of racial organizational theory as a framework brings into sharper relief how emerging literacies can be co-opted for whiteness. As a site for allocating social and material resources, racial organizations draw in not only literacy but literacy as a resource that economically benefits whites the most; they adopt the practices and beliefs about race to dictate literacy distribution within their organization. In doing so, they can constrain and stratify racial groups' agency with literacy from within to support the racial order without. I contend that constraints and stratifications work across

interrelated organizations in the so-called pipeline of computer code bootcamp to software development. Explicitly linking racial organizations with coding literacy tells a complicated story of how the coding movement, the ready-to-work model, participates in and perpetuates racism inside of tech. This helps further challenge aspirations of and claims to a post-racial tech industry as the end-goal of the pipeline rhetoric.

Racial organizational theory and coding literacy help transform emerging literacies' association with Blackness into positive practices. Naming Black coding literacies reveals new alternative ways Black people use coding literacy resources to achieve different outcomes. These changes in "material relations … through human agency" creates conditions that change the racial order overall (Ray, 2019, p. 47). Without giving too much credit to emergent literacies as autonomous forces (Street, 1984; Vieira, 2016), new types of writing may be the tools that activate social justice efforts. I focus on emerging literacies having this possibility because they challenge public schools, post-secondary institutions, and workplaces to create new practices that broaden people's access to these desired literacies, as is happening with more diverse students and adults learning computer programming. It is the moment before they are highly embedded in everyday life and professional practice that emerging literacies can evoke new conversations on societal racial order. What anti-racist possibilities exist for coding literacy education in computer code bootcamps? I show glimpses of an answer later in this chapter and in the Conclusion. In the next section, I explain how racial organizational theory applies to the relationship between computer code bootcamps and tech companies, focusing on hiring practices and education.

Big Tech and Computer Code Bootcamps as Racial Organizations

In Spring 2018, a few months after completing my participant observations at Clearwater Academy, I attended a diversity in tech panel hosted by the downtown public library. Local tech professionals, community activists, and computer science professors spoke about race, racism, and working in Sakowin's technology sector. Among the panelists was Richard, the technical skills instructor for Clearwater Academy. In his assessment of the local job market for Black and Brown coders, Richard noted that tech companies invested in white coders from outside of the state rather than local talent. Few coders of color already worked in Sakowin, and the tech sector, panelists indicated early in the discussion, was hardly welcoming to those historically excluded coders. The insistence on hiring nationwide further exacerbated disparities in hiring diverse coders. The city itself didn't offer a flourishing future for BIPOC

workers and their children, so even if the workplace restructured its culture, the community around it burden coders' families.

When local tech companies did hire, they pulled talent from a Sakowin University, a well-resourced and internationally recognized PWI. One hiring manager on the panel explained, "When you think about a four-year traditional college like [that school] as the only source for talent, you eliminate a lot of people who can't, one, afford the college and, two, who don't want to go to the college because they don't like the [racial] environment that it cultivates." Instead, tech companies should look to other learning environments many BIPOC attend, such as community colleges and Historically Black Colleges and Universities. In addition, panelists noted, there are a host of nonprofit organizations that work exclusively with Black youth interested in tech.

This panel discussion shows how the microcosm of tech in Sakowin carries the influences of macro-level organizations of Big Tech in Silicon Valley. These smaller local startups use Big Tech's racial schemas to determine how coding literacy, and its rewards in the economy, gets distributed across racial groups. It is well-known that Big Tech prides itself on color-evasive racism, post-racialism, and meritocracy as its guiding philosophy for hiring, investment, and technological design. In their far-reaching analysis, however, internet studies scholars Safiya Noble and Sarah Roberts (Noble & Roberts, 2019) find that Big Tech's acceptance of post-racial ideology re-enforces racism. In hiring and representation of its workforce, for example, venture capitalists invest money and resources to startups based on cultural fit, a way to categorize people "most like themselves, or most like the networks they engage" (Noble & Roberts, 2019, p. 119). Those networks may often be white, male, and heteronormative, thus eliding that any Black-owned startup will not receive funding. One may argue that Silicon Valley invests in people from the Global South to indicate their commitment to diversity, but this, Noble and Roberts note, are screens to avoid discussing their consistent discrimination against *domestic* Black, Latine, and Indigenous people. Even the representation of Global South people in the workforce hides racist tropes and stereotypes to suit the needs of their white counterparts, as seen in the television show *Silicon Valley* (Noble & Roberts, 2019).

Big Tech recruits new software developers from the best computer science programs in the United States, the top three of which include University of Washington; University of Berkley, California; and Stanford University (Staley, 2017). These programs have majority white and male recent graduates, many of whom already have internships at top tech companies. Their knowledge of computer programming coupled with learning the cultures of tech companies suggest they are primed for working within the hallowed halls of Big Tech. Meanwhile, organizations that serve Black people—HBCUs, nonprofits, extracurricular clubs, and computer code bootcamps—maybe perceived

as not producing as talented coders as majority-white educational programs. This racialization of coding literacy can then "[replicate] across many organizational forms", "formalized by gatekeepers and exert top-down pressure on subordinates, potentially shifting the relation between schema and resources" (Ray, 2019, p. 34). Richard and Jessica speculated that one reason Sakowin tech companies wanted to support Clearwater Academy was not only the value of computer code bootcamps but also because they were spooked when Google revealed that their workforce was majority white and male in 2014, advocates called them out on their failures to diversify. They were more than happy to collaborate with the only accredited computer code bootcamp that trained historically marginalized people to avoid tough criticism.

Computer code bootcamps can play a crucial role in replicating racial schemas of coding literacy. In doing so, they maintain coding literacy as white property. Recall from Prendergast's analysis above, that granting equal and equitable access to literacy depreciates its value to white people, which can inspire white flight and question the value of education itself. I would argue that because our digital ecosystems rely so much on white perspectives and lived experiences, white flight isn't a viable choice in response to equal and equitable access to computer programming for work; to do so would leave the tools and opportunities of software in new Blacker and Browner hands responsible for our digital ecosystem, one that may challenge whiteness. Computer code bootcamps must *prevent* white flight and the so-called loss of computer programming standards as more Black and Brown coders join the industry.

Computer code bootcamps as racial organizations facilitate insights on race and coding literacy as a practice. For the rest of this chapter, I explore the "material and social-relational foundations, not to mention [the] tangible consequences" (Burke 2016, p. 104) of computer code bootcamps as racial organizations. What's the *impact* on the instructors and their adult learners as they feel the call to uphold standards of computer programming and standards of professionalism in a white context while trying to promote racial justice and Black independence? In exploring this question, I consider how Clearwater Academy instructors and adult learners have begun rethinking the racial schema-resource coupling of coding literacy in computer code bootcamps.

Coding Black Functions for White Software Programs

In the section above, I've explained how Big Tech and computer code bootcamps are interlinked racial organizations based on hiring practices and educational training. Here I want to explain the technical metaphorical framing that rises from my analyzing the focus group interviews and observations I conducted at Clearwater Academy: coding functions. Functions are mini-programs

that contain a set of instructions to accomplish a one specific task. They are easily portable across different kinds of programs, and the coder doesn't need to know what specific lines of code run the function; often coders will "call" or command the computer to run invisible instructions within the function. The more I listened to adult learners and instructors discuss their training in coding literacy, the more I began to re-think Clearwater Academy's racial justice project as a coding sprint itself. Figures 3.1 and 3.2 below lists a series of functions for a banking application I wrote in Python. Each function begins with "def," followed by a descriptive name of the function and the variables that function will take in. In Figure 3.1 the first function is the checking account balance, and it takes in all transactions a client has completed. The function must then do something with that data. In this case, the program includes a mathematical formula—"sum(transactions)"—that will total the number of transactions and then reveal, or "return", the total. Functions like these combine to create the program seen in the second image, which includes user inputs and if-then statements to activate functions. What results is a fully working, though rudimentary, online banking system. The functions in this banking program never change; they, I would say, remain dormant until they are called to carry out specific tasks that make the entire program work.

Writing researchers Jonathan Alexander, Karen Lunsford, and Carl Whithaus (2020; Whithaus et al., 2022) note that scholars use many metaphors to describe their studies of writing (i.e. worlds apart, literacy in the wild, ecologies and networks, and transfer) to "see different aspects of the writing process, the rhetorical situations in which writing occurs, and the perceived agency of writers themselves" (Alexander et al., 2020, p. 106). In addition, the metaphor used "might determine and condition how scholars are approaching, understanding, and analyzing their objects of study" (Alexander et al., 2020, p. 107). The authors wonder how metaphors can do more than just determine approaches to studying but also suggest what hasn't been studied or what implications different metaphors have for writing. I find their longitudinal work on "wayfinding" -- a new metaphor they suggest studying to understand the writing lives of college alumni -- inspirational for focusing my own analysis on what's happening in Clearwater Academy and its operations as a racial organization that codes functions.

```
def balance(client_transactions,name):
    """Adds the total number of transactions the client has completed"""
    return sum(client_transactions)

def available_credit(client_transactions,name):
    """Returns the available credit that the user has left"""
    return customer_accounts[name] - balance(client_transactions,name)
```

Figure 3.1. Screenshot of two functions written in Python.

```
def main():
    """This function runs the electronic banking program."""
    print "Welcome to the Credit Card Accounts Online!"
    print ""
    while True:
                account_limit = random.randint(500,50000)
                print ""
                print "What would you like to do? \n 1. Create a new account \n 2. Make a
                choice = input("Your choice:")
                while choice < 1 or choice > 4: #If user input is less than 1 or more tha
                        print "Invalid choice!"
                        choice = input("Your choice:")
                if choice == 1: #if condition for choosing 1
                        name = raw_input("Account name: ")
                        while name in customer_accounts: #checks if the name is already i
                                print "Error: account", name ,"already exists."
                                name = raw_input("Account name: ")
                        customer_accounts[name] = account_limit
                        client_transactions[name] = []
                        print "New account created for", str(name) + ".", "Credit limit i
                if choice == 2: #if condition for choosing 2
                        while not customer_accounts: #checks if there are no accounts in
                                print "Error: no account exists."
                                print ""
                                main()
                        name = raw_input("Which account? ")
                        new_transaction = input("How much is your transaction? ")
                        while new_transaction < 0:
                                print "Invalid transaction! Must be a positive number."
                                client_transactions[name] = input("How much is your trans
                        if check_availability(new_transaction,client_transactions[name],c
                                make_transaction(new_transaction,client_transactions[name
                                print "Success! Your balance is", str(balance(client_tran
                                print ""
                        if check_availability(new_transaction,client_transactions[name],c
                                print "Transaction rejected. Your available credit is $"
                if choice == 3: #if condition for choosing 3
                        while not customer_accounts: #test if user inputs 3 as a choice w
                                print "Error: no account exists."
                                print ""
                                main()
                                print ""
                        name = raw_input("Which account? ")
                        if name in customer_accounts and client_transactions: #if the nam
                                print statement(client_transactions[name],customer_accoun
                        else:
                                print "Error: no account exists."
                                print ""
                main()
                if choice == 4:
                        print "Done!"
                        quit() #this function exits the program altogether

if __name__ == "__main__":
    main()
```

Figure 3.2. Screenshot of rudimentary banking account system made of multiple functions.

When used as a metaphor, functions in object-oriented programming have different implications for racial hierarchy in the project to democratize coding and diversify the tech industry. Through the method of writing code, mixed with the purpose of computer code bootcamps—job training—adult learners learn to support the profit imperative of Western corporations. To be clear: the economic potential of learning coding matters for Black people, given the tremendous wealth gap between themselves and white people (Shapiro, 2017). Clearwater Academy adult learners in this study had practical economic reasons for attending the computer code bootcamp, such as becoming

entrepreneurs to support their families and local communities. But the way code works for Black people specifically—a cultural basis for computer programming—isn't the main concern for some computer code bootcamps. Eliding a Black coding literacy suggests that computer code bootcamps train Black people as small programs that can be reused in one company after another without disrupting the industry's mandate to preserve white technological design. They are coding Black functions.

As instructors Richard and Jessica and the Black adult learners discuss below, the purpose of Black professionalization into software isn't to create dynamic, creative coders but rather docile and passive members that bring color to tech while keeping its sociocultural practices. While "cultural fit" matters for promoting community-building and inclusion, it can come at the expense of promoting the health and well-being of its marginalized literate workers. Cloaked with credentialed whiteness, these Black coders are functions meant to "disappear" or "fuse" into the racial organization, continuing the race systems that perpetuate inequality.

In the following section, I describe Clearwater Academy's teaching and assessment practices and the challenges of teaching coding literacy and employability skills. First, I explain the exigencies that shape Clearwater Academy's curriculum design. Then I explain in-depth how Richard and Jessica teach this curriculum. Their reflections and intentions reveal critical awareness that helping low-income racially marginalized adults succeed in a majority-white tech sector requires that they shape them into literate subjects that may exclude their full humanity as Black people. I weave into this section perspectives of the adult learners to show the conflict between Richard and Jessica's honest hope for their adult learners and the expectations of their Black adult learners. My analysis postulates that Clearwater Academy has been positioned to materially and metaphorically code Black functions for white software programs, but my analysis also attempts to reconcile the two perspectives on the material consequences of working in computer code bootcamps as racial organizations.

We Are Not a School: Exigency for Clearwater Academy's Curriculum

When I first visited Clearwater Academy in Spring 2017, I expected significant emphasis on learning HTML, CSS, and JavaScript. Richard and Jessica split the curriculum at that time between 50 percent coding instruction and 50 percent "employability skills" instruction, which included learning the written genres and behaviors associated with the workplace and professionalism: mock interviews, résumé and cover letter writing, timeliness, self-care, and financial well-being. By fall 2017, when I returned for my second round of participant

observations and interviews, Richard and Jessica increased employability skills to two-thirds of the curriculum. Why de-emphasize computer programming itself? As I describe in Chapter 1, the professional and cultural backgrounds their adult learners would bring with them weren't suitable for tech. "They have to get in a room and work the room, and they've never been in a scenario like that," Richard explained to me during an interview in November. "In all honesty, the bottom line is without the employability skills there is no job for these folks. There's no way they get into these positions—in briefcase, suit-wearing business-type positions—after having gas station and retail being the best opportunity they ever had. If we don't teach those skills, they cannot survive, and they will blow it for everyone following. It is that simple."

Richard's observation evokes a persistent problem during Clearwater Academy's early days. In the first two years of the training program, Richard's predecessors, two Black women coders, spent most of the 14 weeks teaching adult learners web design and computer programming. However, graduates interning with tech sponsors and other interested employers had trouble adapting to the companies' "appropriate ways of acting, interacting, participating, and participant structures" (Gee, 2011, p. 90). The following quote from Chapter 1 relates to my analysis in this chapter: when Richard spoke with tech sponsors about their experiences with Clearwater Academy, they told him that "there's no way these people should be in our office working. These people didn't know how to work, be employees. They had very bad habits. Social skills."

To get an idea what these "bad habits" looked like, Jessica showed me an etiquette quiz that she used to prepare adult learners for work. Each question was based on actual incidents in previous internships. By the time I began my study, Jessica had stopped using this quiz, but it nevertheless demonstrates the social and cultural tensions between Clearwater Academy graduates and its tech sponsors. In the directions, adult learners are asked to give their "honest answer [on] how you would act in the work environment." For example:

> 2. While you're waiting for your computer to be fixed …
>
> a. It'll take a while. Kick back, relax and take a quick nap.
>
> b. Pull out the employee handbook, prop your feet on the desk, and read it.
>
> c. Find your supervisor and ask them what you should do in the meantime.
>
> d. Ask your coworker if they need help.

Other incidents the quiz covered included negotiating salary, how often to take a smoke break (the first choice is "8 – one every hour") and having

a consistent work schedule. Interns made fried chicken in breakrooms and tried to sell marijuana to their supervisors while on the job. The "social skills" problem occurred even before the internship. Richard and Jessica scheduled tours of tech companies for their adult learners throughout the semester, so they could get first-hand experience of what it's like working in a tech office. During a visit to one major tech company in Sakowin, Richard saw adult learners "[clean] out about fifteen bowls of candy distributed throughout fifteen rooms." While the company did allow them to take some candy, that did not mean they could "take handfuls and put [them] in their pocket," Jessica explained. Richard framed the incident as a "Rodney King Loot Night," referring to the 1992 Los Angeles race riots that occurred after a jury acquitted four white police officers of excessively beating Rodney King. Between April 29 and May 4 rioters looted many stores owned by Asian/Asian Americans and Black people. Learning from this and other incidents, Jessica would tell future adult learners that they "take nothing" from these companies. A few semesters later, one class followed this direction: although a company representative said they could take popcorn and water bottles from the boardroom, no one took anything until Jessica gave them permission. "It was so, so good," Jessica said, "because they had taken us seriously about these things. And I think it just shows a different standard, even compared to [the fall 2016 class]. It's good to be taken seriously but it's also because they're learning."

While adult learners learn professionalism on one side of the curriculum, instruction in web design (HTML, CSS, JavaScript, and Wordpress) follows Google's learning about learning framework on the other side. Using FreeCodeCamp to learn HTML, CSS, and JavaScript taught adult learners the basics in web design. The website also inadvertently demonstrated the importance of following directions in the workplace. When adult learners understood the rudiments of web design, Richard became interested in their "[being] able to prove to me that you can read, follow instructions, make it through the steps, because that's what any job is about. You're gonna be given instruction, given a task, and it's gonna be up to you to go from here to here. My biggest focus is teaching them that, not everything about JavaScript." Learning coding itself wasn't an issue, because the nature of computer programming awarded access to more types of languages. As Richard explained during our interview in the fall, "Once you have the understanding of how programming works and their logical processes, then you can learn any other programming language in seven days." Adult learners would have to juggle multiple languages to get the jobs they wanted until one day they would only need to become, in Richard's words, a "guru" in one language later in their career.

I have experienced what Richard describes. The semester before I began my ethnographic research at Clearwater Academy, I took a college course in Python,

a computer programming language considered the "gateway" to computer science. A year later I tried helping Zeus with an exercise about variables and if-then statements in JavaScript. Although I did not know JavaScript, I could easily identify which lines of code had variables. I pointed these out to Zeus and explained their purposes. With my help, he finished the exercise and moved on to the next. Richard made sure that adult learners work on code independently in FreeCodeCamp but take those skills into team projects. They prepare and practice mock interviews and elevator pitches with mentors from the tech industry, and they work through coding problems with tutors each week.

Richard and Jessica made an important distinction during both interviews on Clearwater Academy's adult learners, curriculum, instruction, and assessment. "The bottom-line is this is employment training, not a school," Richard explained. Jessica co-signed this point in a separate interview, emphasizing that "we want to create employable people, and that is what we care for." While higher education is often mandated to make adult learners career-ready, they still do so under a model that distributes knowledge about computer science for a degree that prepare adult learners for a variety of other career pathways; Clearwater Academy is in the business of coding literacy education *for work*, whether adult learners become freelance coders or employed coders of a company. Clearwater Academy has a specific environment in mind, and their teaching coding literacy in the sociocultural contexts of offices and clients will get Black adult learners there.

The move from teaching only web design to teaching coding literacy in its context aligns with theories in New Literacy Studies. For example, the initial strict focus on coding in the early days of Clearwater Academy suggests that coding alone will make adult learners successful in the same ways anthropologists once postulated that ancient literate societies modernized because they developed writing, which shaped the mind to closely analyze and question the world around them (Olson, 1977; Ong, 1986) Literacy scholar Brian Street(1984) argues that this perspective (an autonomous model of literacy) gives too much power to literacy and ignores the ways literacy actually helps *create* and *support* existing inequalities and ideologies that accumulate institutional power for some and stratification for everyone. Understanding literacy as a social practice (Barton & Hamilton, 1998) in its variety reveals power, oppression, and liberation in the complex ways these three forces interact for human beings. Language use, ways of being, and ways of doing coalesce into communities and social institutions that people must fit into. Coding literacy facilitates these interactions as much as communities of practices defines the coding literacy practices that matter (Wenger, 1998).

Too much emphasis leaves adult learners floundering and failing in the sea of values, traditions, and expectations of being in the tech workplace; like the

autonomous model of literacy, the context in which coding works goes unnoticed. In practice, computer programming and employability skills *can't* be taught separately. Richard and Jessica recognize the conflict and teach Black adult learners to consider that communicating technical knowledge with clients and employers—in writing and speaking—are key components of their coding literacy (i.e. learning how to talk to clients does require knowing how web design works but Clearwater Academy adult learners must learn how to be personable with those clients). Under an implicit racist system, Richard and Jessica cannot draw connections between coding and the sociocultural and material contexts of software development that help Black adult learners imagine other possibilities. The curriculum design defines diversity and inclusion according to the tech sponsors' definition (see Chapter 2 for more) and not in ways that change the companies' cultures and practices.

In the following pages, I first describe the assessment philosophy Richard and Jessica deployed in Clearwater Academy. Then I dive into specific examples that demonstrate how their assessment practices can be racially coded in ways they didn't intend but nevertheless they find necessary. Finally, I show how they and Black adult learners observe that these racially coded assessment practices suggest the real intentions of the tech sponsors interested in Clearwater Academy. The materials and social consequences, they find, do not advance them forward materially or socially but rather create what I call Black functions for white software programs. However, these conflicts between figured worlds (Gee, 2011) open pathways of escape from being a coded Black function.

Racially Coded Assessment Practices

Earlier in this chapter I explained how Clearwater Academy wasn't a school; it was a training academy for employability in software development. To underscore this point, the class syllabus announced to adult learners that the coding bootcamp was a "worksite." Adult learners were "employees" subjected to all the expectations that come with that title: attendance, turning in work on time, and following instructions. Richard and Jessica hoped that their adult learners would take the training program seriously. The point was to show adult learners "what it's like on the job." Turning in assignments late, arriving at Clearwater Academy late, not attending "class," and missing mandatory meetings like mock interviews and company tours were the top reasons adult learners failed Clearwater Academy, not their computer programming skills. In fact, behaving like employees can matter more than the content they produce. Jessica explained that a good résumé and cover letter matters but adult learners received points based on simply turning in assignments on time.

Even the best coders must be held accountable for failing these employability skills. "It's a complete disservice if we let them through," explained Richard. "When people aren't held accountable and given the opportunity, they don't know what that opportunity offers, what the value is, what it really means to have it, because you haven't gone through the process." The stakes could be high for everyone at Clearwater Academy: failing an internship opportunity because of their behavior may permanently ruin that partnership for Clearwater Academy. Richard and Jessica were wary to recommend a graduate for hire who hadn't demonstrated honesty, integrity, and accountability.

That said, Richard and Jessica did not let the "objectivity" of points and attendance drive assessment; they carefully attended to the subjectivity of their relationships with adult learners. The instructors recognized that they were working with people and that they truly desired that everyone succeed. Assessment was highly individualized based on their adult learners' needs. The instructors spent the first three weeks getting to know their adult learners to determine how best to communicate and assist in their success. Richard and Jessica balanced blunt feedback on adult learners' performance with real interest in supporting their learning and well-being. The tone and type of feedback shifted from person to person rather than everyone getting the same feedback. The instructors thought this individualized approach to assessment equitable. Because Clearwater Academy considers itself using a holistic approach to teaching and assessment, Richard and Jessica were aware of the racial and gender dynamics of their relationship with adult learners. Richard was a Black man and Jessica was a younger white woman. When they needed to pull adult learners aside for tough talks, Richard and Jessica both discussed who should lead the conversation based on the race and gender of the adult learner and the topic, especially if the topic had implications for one or more of the adult learners' social identities.

For example, in Richard and Jessica's judgement, Zeus, whom we met in the Introduction and in Chapter 2, needed tough love. In the beginning of the semester, Jessica said he "drove us crazy." At one point, any wrong move would've expelled him from Clearwater Academy. Halfway through the semester Zeus had been falling asleep in the middle of lectures and had a bad attitude with instructors and classmates. Richard and Jessica spoke with Zeus privately to figure out what was going on. He walked into the office "stone cold, like did not want to talk to us," explained Jessica. Both instructors "said some things that were harsh" but followed that up with "lots of love" that encouraged Zeus in his success. Zeus was almost in tears after the conversation. "His willingness to change," explained Richard, "is what keeps him in the class." Richard and Jessica speak truth to adult learners so they can prosper, which is why every conversation—no matter how blunt—ends with "What

do you need for support from us now? What do you need from us?" Richard and Jessica's assessment practices—as they emphasize again and again in our interviews—tied directly to the workplace itself. Tough conversations with adult learners about their behavior in class prepared them for what will happen; they will not, as Richard explained, "sugar coat it or make me think I'm walking into a fantasyland when you're basically walking into hell."

So far in reviewing assessment practices, Richard and Jessica prepare adult learners for the sociocultural contexts in which coding happens. There are a variety of implications for how timelines and collaboration matter in a company responsible for offering digital services from hundreds to millions of people. From the technical skills side, a small section of code can break the Internet (Collins, 2016; MacDonald, 2018; Williams, 2016); but failing to deliver a software package on time equally disrupts companies and their users' digital lives; a freelance developer late on website design doesn't get paid by their client on time, if they get paid at all. Showing up and not making excuses, for Richard and Jessica, makes computer programming and software work for users.

But Clearwater Academy wasn't just teaching a series of behaviors and habits that help adult learners leverage coding literacy into financial security; the assessment and teaching Richard and Jessica used also *shaped their adult learners' identities*. One of the interesting lessons Richard gave during my participant observations was the rhetorical work Black adult learners must do to get a job. Using himself as an example, Richard lectured to adult learners about how he didn't know every computer programming language, framework, or library, nor did he need to know them to get a job or get a new client. All Richard needed to do was walk into the room and make the employer or client *believe* he can do the work; if Clearwater Academy adult learners can make people believe they can do anything, they could be successful. During our interviews on assessment, Richard explained that he was hired to do one thing: "How to be whoever you need to be to get the job done."

Richard and Jessica understood they were in a difficult position on shaping adult learners' identities for computer programming. As Clearwater Academy became more well-known, an increasingly diverse group of adult learners applied to the training, partly thanks to graduating adult learners recommending the computer code bootcamp to their friends and family. Encountering new cultures and linguistic practices, Richard and Jessica begrudgingly accepted they participated in the racial order's demand that marginalized people assimilate, that they had to promote in veiled and not so veiled ways anti-Black linguistic racism (Baker-Bell, 2020). "Culturally and language-wise," Richard explained, these adult learners needed further teaching. One adult learner was a brilliant architect with big picture ideas, but Jessica could never

understand what he was talking about; another adult learner's job materials attracted plenty of requests for interviews but not one of those interviews landed the adult learner a job. The problem wasn't her English; employers considered her accent too thick. Richard and Jessica had one of the best adult learners in their class during my visits: a Black woman from Chicago who was "Super bright, good at asking questions, on top of things. Whenever she needs help, she goes to Richard's desk and waits patiently," explained Jessica. But her linguistic practices, learned from growing up in Chicago, hid her potential to employers; Jessica volunteered to help work on her speech. She admitted that there is nothing culturally or linguistically wrong with this Black woman adult learner, nor any other adult learner in Clearwater Academy. "It's just some of those barriers that employers, whether for better or worse, don't like, and will hold that against her. It's really not fair. it's hard for me as a white person because I ask people, 'Come to my side.' And it feels like that sometimes. It's not fair and I hate it. But at the same time we have to make decisions. Do you want a job, or do you not want a job?"

Individualized assessment and humanized relationship-building creates a culture of care in Clearwater Academy, yet they serve to fulfill a capitalist and racial outcome: create an employable person in tech, to teach them to be what they need to be to match the needs of their client or employer. Even if that means learning characteristics associated with whiteness. Humanizing assessment achieves that pragmatic philosophy of employability based on the conditions of whoever is convinced to hire Black coders. Under this pragmatic philosophy, computer code bootcamps can train non-resistant, docile coders (taking on any function name that defines their role in the program) that receive tasks (take in data), perform the tasks (execute instructions about that data), and produce a satisfying result (return some value). In my participant observations and focus interviews with instructors, there's no room for critiquing or questioning how the software or websites they design for others prompt social good or how to protect themselves from cultures of whiteness. Despite a program focused on training the United States' most vulnerable people, Clearwater Academy cannot position themselves to train an ethical or justice-informed technical communicator who can draw on their lived experiences and training to affect decision-making, from the smallest client to the largest company. A support assessment model would emphasize the Blackness missing from technological design and culture. Instead, careful emphasis on honesty, integrity, and good work helps Clearwater Academy re-write some lines of code and delete other lines of code in Black adult learners to make them functional pieces of white software design, exorcizing Blackness. Professionalism, capitalism, and pragmatism, nestled within humanizing assessment, teaches Black adult learners the cultural commands of whiteness

in software, how to plug in and stay in. This model works best for Clearwater Academy because it's the only model that seems to be acceptable to the companies responsible for Black social mobility.

The re-programming project on the bodies of Black people requires that material and social resources focus on adult learners most capable of becoming Black functions. The benchmark for success is professional whiteness, and the best opportunities continue to draw toward those adult learners. We've seen this play out with Zeus whose life circumstances and own perspectives on Clearwater Academy nearly forced Richard and Jessica to dismiss him from the class; in a hierarchy of who gets the resources for success, Zeus, for a time, was at the bottom. Interviews with the instructors revealed other instances of carefully directing resources where necessary. A recommendation from either or both Richard and Jessica lend powerful social capital to a Clearwater Academy graduate: internships and employment opportunities abound. Not all adult learners get these recommendations, however, despite their completing the training. They can return to Clearwater Academy for help the next semester and for the rest of their lives, even—after all, the program is still connected to a larger non-profit invested in social justice—but some opportunity has been missed because of their work during the program. Clearwater Academy, as an example of racial organization, commits to carefully distributing resources to adult learners with better circumstances and learning strategies than others, creating a gatekeeping mechanism that releases the most workable functions to the world and helping others find their own way to social mobility. Resources move based on who masters (pun intended) white racial schemas and those who do not delete their lines of Blackness.

"The New Shiny POC"

I conducted focus group interviews with participants from the 2017 spring and fall classes at Clearwater Academy in small, private rooms. I completed three interviews with the spring semester's adult learners at the beginning, middle, and end of the semester to capture their shifting perspectives, if any, as they worked throughout the program. Scheduling conflicts resulted in one end-of-the-semester focus group interview with participants from the fall class. Despite having only one focus group interview with the fall 2017 class, similar themes appeared in both classes. Much of what I describe and analyze below come from those final interviews and my participant observations.

Like Richard and Jessica, Black adult learners noted the difficult relationship among themselves, Clearwater Academy, and majority-white tech companies, sponsors, and other interested employers. Black adult learners identified the consequences of overemphasizing teaching the sociocultural contexts

of the tech workplace without challenging those cultures or showing adult learners how to use their lived experiences to change or create racially just coding cultures; they had joined a social reproduction (Collins, 2009) project that supported the practices of racial organizations rather than a social good project that benefitted them, if they could get a job in the first place. Black adult learners in this study grappled with the costs and benefits of this approach to acquiring coding literacy.

For example, Clearwater Academy's growing reputation attracted more support from the local sector over the years. During the spring 2017 focus group interview, Alice thought tech companies delivering money and other resources to computer code bootcamps like Clearwater Academy was a sign of goodwill toward anti-racism: "If they didn't support the mission, they wouldn't even partner with [Clearwater], they wouldn't take interns, they wouldn't take employees, they wouldn't donate money, they wouldn't do any of that." She wanted to stay optimistic that tech companies didn't just want to "look good." If that were the case, Alice reasoned, they could just hire one Black person and be done. With so many resources going into training low-income racially marginalized people and women, Alice believed, there must be something more than that—they must care about Black people developing coding literacy practices. However, other Black adult learners wondered if the tech companies' investment in Clearwater Academy really helped diversify the tech industry and promote anti-racist practices within coding. Alex, a classmate of Alice's, did not think that directing resources and money in their direction aligned with her optimism:

> I have seen people -- *I have seen people* -- who will do things publicly that are real good just for the sake of getting the kudos of having done something good … If you see, "Oh now I feel bad. Let's grab one or two and let's just make us feel better." Or "Let's grab a whole bunch and make sure everybody knows we feel better but really we still feel the same way that we feel."

Alex accused tech companies of being what law professor Nancy Leong calls identity capitalists—"ingroup members who profit from outgroup identit[ies]" (2021, p. 3). That is, members of dominant social groups seeking out cultural and social values from marginalized people using a variety of practices from hiring to citing a marginalized person's support of dominant social groups' behaviors and policies. Identity capitalists make superficial gestures toward diversity, equity, and inclusion for financial and cultural capital without causing real meaningful change in the systems that oppress marginalized people. Identity capitalism is reminiscent of the operations of racial

organizations who deliver just enough resources according to racial schemas to satisfy diversity, inclusion, and equity advocates, or at least counter criticism from diversity advocates. I addressed this idea briefly earlier in the chapter. For Alex, tech sponsors only deliver resources to Clearwater Academy so they can either exploit the coding literacies and identities of the adult learners they do hire or add Clearwater Academy to a list of initiatives that show their commitment to diversity.

Although they questioned the depth of Clearwater Academy's sponsorship, Black adult learners nevertheless wearily accepted the necessity of turning to majority-white tech companies for support: they possessed the best resources and wealth in town. Kevin, another participant in the spring 2017 group interview, acknowledged this reality when his responses combined Alice and Alex's perspectives. But he later considered the consequences this relationship had on them as adult learners: "I like what they're kinda doing," Kevin explained to his peers, "but I hate the fact that they gotta show me off to people. That they're like 'Hey, look at this new Shiny POC who can do all this stuff. Come look at him! Come look at him! He's nice and strong! He knows his code!' You know what I mean? I hate that stuff." Kevin's imaginative take on presenting the Black coder to white developers recalls images of enslaved Black people standing on auction blocks for white slave masters in the antebellum South. While on display, the seller—in this case Clearwater Academy—names a list of computer programming languages they know and the various programs or websites they've built. Tech companies then bid on who they want on their tech plantations. That language is harsh, but it nevertheless reflects Jessica's point about getting a job according to white standards, which means following the social engineering that removes many Black behaviors that wouldn't be desirable to white employers. The language—tech plantation—also brings Kevin's metaphor to its logical conclusion. What Kevin calls the Shiny POC is the Shiny Black Function, the outcome of Clearwater Academy's curriculum and assessment, even though that is not the intended philosophy for training racially marginalized people in software development.

Another critique adult learners launched against the social reproduction project at Clearwater Academy was that they found they weren't learning computer programming in a dynamic, sophisticated way; that had serious implications for their prospects on the job market. Above, I explained that functions take in data and perform some act; functions *can be* complex, such as taking in other functions to perform actions or containing many lines of code to calculate problems. Kevin jeered earlier that the Shiny POC knows their code but based on the computer code bootcamp's curriculum and assessment model, the Black function has none of these complex features. The Black function comes packaged with simple lines of code to execute for white

software programs, leaving undisrupted its contributions to the racial system tech can serve.

Black adult learners had high hopes for the program at the beginning of the semester. Rania had learned graphic design and HTML years ago in Atlanta, so coming to Clearwater was like "starting from scratch." She had hoped Clearwater Academy would be "a great opportunity, a new start, … so when I read about it, I said, 'Oh this is gonna be good.'" But once Rania got deeper into the semester, she found learning programming challenging mostly because she and her peers had to work on their coding without much direct instruction. "You really have to practice every day to really understand it and get it. You really have to practice and study because if you don't, you're gonna feel lost every day." Although Rania struggled with exercises in FreeCodeCamp, other Black adult learners found the work unsatisfying and not teaching them enough. "FreeCodeCamp doesn't teach you programming language," Alex, from the Spring 2017 class, explained. "It teaches you how to follow instructions and decode instructions in order to get past those problems." Kevin likewise found Code Academy too rudimentary: "It was the dumbest thing in the world; you didn't learn anything, because it was like 'This is how you use it. This is what it's used for. And this is what we want you to do with it.' So you just copy. Paste. Change one or two letters or words. And it's correct."

However, by the time they started doing exercises in Python and JavaScript, the challenge and pace shifted: FreeCodeCamp, for example, gave adult learners an example but no further guidance. *Less guidance* in problem-solving probably aligned more with what Alex thought was the correct way to learn coding: "Coding, writing code, teaches you how to code. You gotta write code that does shit and you gotta write code that fixes it. That teaches you how to code." What he and his peers wanted to learn was not what a karate chop looks like, but rather "here's how to do a karate chop; here's where you use a karate chop; this is why you use a karate chop." To become better coders, adult learners had to take their own initiative, finding new ways to use what they learned in FreeCodeCamp and Code Academy, or sometimes learning a separate computer programming language on their own, such as Kevin looking into Apple's Swift for iOS programming and Rania using YouTube to expand her knowledge of HTML and CSS. But that's what Richard *wanted* adult learners to do: learn how to learn, use HTML and CSS as a gateway into other computer programming languages. But this knowledge wasn't as intuitive to some Black adult learners as to others.

In the fall semester focus group interviews, Pierre, the most vocal participant, understood where Richard was coming from; he agreed that they must learn and practice computer programming on their own time or "you will feel lost." But Pierre said he felt "owed instruction days [because] I spend so much

time. I wish there was more of a teaching aspect like 'Hey, this is how arrays work …'" Direct instruction would provide him with other benefits, too: While Pierre did practice and study web programming through FreeCodeCamp, he was not sure how well he was doing because instructors did not go over the exercises in class. Without consistent feedback during the class, Pierre only felt more wary of his ability to do well as a software developer. Myra, also a participant from the Fall class, expected direct instruction because they were physically in the classroom to learn web design; if YouTube alone was the teacher, why attend Clearwater Academy? To underlie the importance of direct instruction in computer programming, adult learners in the fall class agreed that their favorite day was Tech Tutor Day: a day when professional coders would visit the class and work one-on-one with adult learners on their coding exercises in FreeCodeCamp. "We need to be spending time doing the technical stuff," emphasized Pierre, but in-class time is instead spent doing "activities, résumé stuff."

Pierre wasn't exaggerating. During my participant observations, I saw adult learners spend up to three weeks on learning how to write effective résumés and doing icebreakers that, to Pierre, felt like high school work. In the words of Alex, from the spring 2017 class, they came for computer programming and instead "got a lot of bonus stuff. Five percent of the class was useful; I did not sign up for employment training or this structured approach to teaching; showing up to a place like it's a job and you're not getting paid for it and taking a lot of grief while you are here."

The learn how to learn pedagogical strategy left adult learners unsure that they could do well when applying to jobs after graduating. As an example, adult learners in the Fall semester class mentioned playing a Jeopardy-style game so Richard and Jessica could test their coding knowledge. The activity came ahead of mock technical interviews with recruiters when adult learners would answer questions and complete exercises to show their knowledge of web design. The adult learners failed the test; no one knew anything despite practicing coding exercises online. Richard was embarrassed; the adult learners would reflect badly on Clearwater Academy when they began mock technical interviews. They seemed to have spent too much time on appropriate body language and eye contact and trying to learn how to talk in a corporate setting. They didn't learn enough, but they, as Myra explained in our group interviews, "can give you a whole rundown on how to talk."

Not knowing code placed a new burden on Black adult learners at Clearwater Academy. Pierre was the most explicit about this problem in his class' focus group interviews. Ideally, he would like to say, "Hey, hey, I can do the same things in the same way you do them and I'm a different color skin." This looks like leveling the playing field for Black people. But the opposite seemed

more likely after graduating: "Hey, I'm a different skin [color] and I don't talk like the rest of the [white] people assume that I speak. Yet I don't know [coding] but I can talk with you … It's intimidating being a person of color … I'm a person of color *and* I don't know shit." In other words, it seemed that rather than being a professional Black coder prepared for working in tech, Clearwater Academy would confirm in some instances the racial schema that Black people do not know code or cannot learn code; recruiting Black coding literacies seems like a bad investment after all. Only some exceptional Black adult learners—Shiny POCs—should accrue the resources of tech for financial well-being under the rubric of whiteness.

While Pierre felt the weight of racism and coding, there were hints of aspiring to be a Black function: doing as well as white coders with only difference being a Black person still suggests respecting coding literacy as white property. To grasp it and hold it, you must not cheapen its quality; you must honor it the way white coders honor it. Aspiring to this standard implicitly remains consistent from the spring class of participants, who tried to push back against Alex's suspicion of the tech company's motives. For example, Alex drilled on the point that they had only learned a skill, and a skill on its own could not topple institutional racism: the problem of racism is too gigantic for tech to solve. Isaiah, however, thought that how tech sponsors felt about training and hiring Black coders didn't matter: "If you can show them your skill with coding, people can't say anything against you if you're good … You just killed them dead. They can feel any type of way they want [about diversity and inclusion], they cannot say 'Oh you are horrible at this. I still feel the same way.'"

Isaiah tried to reconcile what they saw happening in Clearwater Academy by appealing to the merit of coding literacy practice alone. But this appeal suggested their becoming Black functions again: one who is good at coding according to the expectations of the tech culture. While the skill itself may show the Black coder an asset, the employer still had a docile coder who did well but left racial schemas and ways of distributing technology to the public according to white frameworks uninterrupted. True to racial organization theory described above, this kind of Clearwater graduate won't demand that the tech culture needs to change.

But the relationship between Clearwater Academy and tech sponsors is uneven; the power dynamics favors tech companies and their notions of a good employee. Because they have the resources and capital (and job positions), tech sponsors have the power to shape the priorities of a computer code bootcamp's approach to race and racism and how coding literacy works towards those ends, not vice versa. When the curriculum and assessment of Clearwater Academy draws on the resources and sociocultural expectations

of tech companies, the computer code bootcamp programs Black coders who uphold whiteness and coding literacy as white property. They learn this orientation through employability skills but also in computer programming where they follow instructions as given on the screen. Learning computer programming beyond HTML and CSS is encouraged and expected, but the behaviors matter most.

The point of interest here is how Black adult learners and equity-minded instructors feel the weight of the tech industry's racial schema as they use their resources; they understand how racial systems fuse with the project of Clearwater Academy and they wrestle with this program while desiring to break this philosophy of coding literacy learning, even as they must go through that program to succeed. What would it take to rewrite the process of programming Black functions and revise the structure of computer code bootcamps as racial organizations? I take up this question in the final section.

"From a Ripple Effect to a Wave": Advocating for Black Coding Literacy

The public discourse that drives democratizing computer programming education for work hinges on the logic that few Black people work as software developers in the companies that have created and shaped the infrastructure of our digital lives. Coding literacy may help address the legacy of poverty and social exclusion for Black communities. Computer code bootcamps and their relationships with the tech industry can perpetuate the problems the coding movement intends to address, as demonstrated in Black adult learners' and instructors' conflicting reflections on curriculum and assessment practices in Clearwater Academy. Addressing poverty and diversity, equity, and inclusionin tech requires a direct attack on the redundancies of white supremacy. In my focus group interviews, Black participants cared little for these macro-level issues and outcomes. When reflecting on their participation in the coding movement, adult learners instead imagined personal outcomes, what I consider a working vision for Black coding literacy as transformative access (Banks, 2006).

First, some participants acknowledged the realities of going into tech as "early adopters" of the computer code bootcamp model; they were not the cohorts that would witness racism end. Even though Richard and Jessica did not always explicitly discuss racism, the adult learners knew from lived experience what they would be getting into if they did work in software. Speaking in the spring semester, Kevin expected "lots of awkwardness. There's not going to be many people of color [when I get a tech job]. I'm going to jump into a pool of people who aren't gonna want me there. Just dealing with that,

I'm mentally ready to deal with that." Zeus, speaking for himself in Fall 2017, thought he would feel "eyes on the back of my head. People hovering over me because I'm Mexican, and they know that I'm coming with only HTML and JS. It sucks. But more fighting against discrimination. I feel like it's not even direct discrimination—that indirect indiscrimination." Kiesha had the dual problem of racism and sexism, as she witnessed firsthand from visiting one of several Meetups that adult learners could attend to get points. "I don't hear about breaking ceilings at the meetups," she explained. She referenced meeting a woman developer who had 15 years of experience at her company. She fit in just fine and was nice, from what Myra could tell. But this coder never went up the ladder. And she was white. Feeling the weight of race, as well, Myra assessed her prospects in this way: "I look like nobody in the tech field; I have long nails, but I can type my ass off; I have weave down to my back but that don't determine who I am." And, yet Myra could not escape the feeling that her appearance—and lack of knowledge of some computer programming languages—would determine her success. Or failure.

Alex summarized the above thoughts eloquently:

> If you teach a man to fish, you have taught him a skill that is going to increase his chances of survivability. But it doesn't give him privileges like white people. Does learning this skill help me or the African American community? No. I could have learned any other skill that would provide a means to make a living. The mass distribution of this technology impacting African Americans? Probably not. It wasn't one thing that caused how we see African Americans. We need more than just coding to see the impact. Just because you learn how to code, and interview well doesn't mean you know how to placate white people.

Racism has too many moving parts for coding literacy alone to fix. They can dress up as Black functions as much as they want, but they don't expect the cultures of tech to be any different from what they find elsewhere.

However, participants did see their learning at Clearwater Academy the starting point for a macro-level trend toward equity and anti-racism in tech. What came to light in the spring class interviews especially was the notion that so few Black people are seen as software developers. Participants in that group agreed that they were "outliers. We are not the typical [Black] people," according to Alex. For them, Black people have not always shown interest in understanding how tech works. While Black communities are happy to use digital technologies, participants thought their own people didn't have the stomach for building those technologies because coding looked too

complicated. Everyone seemed to have a story about meeting a Black person who seemed technophobic. Consider this exchange: Alex recalled hearing "about a mother from the last cohort who tried to get her daughter into this [Spring 2017] cohort. She's a Black girl; she's not technology inclined. She said that 'computers are smarter than me.'"

DeAndre chimed in, "If that ain't about the dumbest——you control the computer, the computer is about as smart as you are."

"Not for our people," replied Alex. "I think our people think the computer is smarter than they are because they don't … they don't know that the computer only says yes or no. True or false. Zeroes and ones."

On one hand, I find this perspective misaligned with other accounts of marginalized people's relationship with technology. Political scientist Virginia Eubanks (2011) in *Digital Dead End: Fighting for Social Justice in the Information Age* worked with diverse low-income women at a YWCA technology literacy program; women in this group, including Black women, were hyper-aware of how city governments that deployed technology to help the impoverished were also methods of surveilling those same poor people. And historian Charlton D. McIlwain (2020) documents decades of Black engagement with technological systems, as well. I don't, and cannot, however, discount the lived experiences of Black participants in these interviews. Within their local communities, in their relationships with other Black people, they consistently found reluctance among the young and old to participate in the designs of software.

Becoming the first of a few Black people learning coding literacy still set them as role models for their community. Rosie observed that many trailblazers had come before them; everyone in the spring semester class was joining in that rich tradition of trailblazing Black people. With coding, Rosie explained, they can "leave a mark in the world … Everyone at this table has the opportunity to make a change in this world." Clearwater Academy's challenging racism and poverty had to be long-term and ongoing, but the Black adult learners in this study could at least be "a ripple effect," and with more Black people entering tech through computer code bootcamps they could later become "a wave," according to Rosie. However, it's not just themselves they considered making that wave. Clearwater Academy also had some responsibility in making this wave possible. The computer code bootcamp must rely on majority-white tech companies because, as explained before, they have power and resources to mete out how coding literacy gets taught to Black adult learners. Kevin hoped that Clearwater Academy would later seek partnerships with the admittedly less powerful Black-owned tech companies (once again evoking the principle that Black-owned organizations do not have the same position in a racial organization system) and no longer rely

on the money and resources of their tech sponsors.

Participants seemed to have less interest in fixing tech itself at this point in the history of mass coding literacy, and more interest in using coding literacy to better themselves and their community. This was the answer to Alex's point made in the previous section: yes, there is no white privilege with computer programming, but the work here isn't about white people in tech. As Rosie clarified to Alex, and as Isiah agreed with her, "It's about what *you do with the information you have*, and you don't want to be your own worst enemy. … You just gotta keep on pushing past that … What are you going to bring for you and your family?" Others did ruminate on the possibilities after graduating from Clearwater Academy, goals that became increasingly local and personal to their individual context.

What would Rosie do with the information and resources she gathered from Clearwater Academy? As a fifty-eight-year-old Black woman in early retirement, Rosie thought she could later become a bridge between elderly Black people like herself and the technologies they use, or don't use. Having decided to be a freelance web developer, Rosie followed Richard and Jessica's suggestion to think of three people or organizations she could do business with. This idea seemed necessary, especially as she knew many Black-owned businesses did not have a website to reach a wider audience. But more than her own financial benefits, Rose wanted to share her "knowledge to help other people that maybe in my age group—you know, retired or about to or whatever—can utilize these same tools to help them do some of the same [things]." Similarly, Kevin envisioned himself bringing what he had learned back to Arizona where he volunteered with a Black Lives Matter chapter and an LGBTQ+ advocacy nonprofit. "It may not make the hugest impact but increasing the amount of knowledge that a person knows is infinite," explained Kevin. "Because then they can teach their people, they can teach their people, they can teach their kids. I feel like my personal ripple is just helping my community and my kids." As for leveraging computer programming into social mobility, Kevin thought of collaborating with his Blerd (Black nerds) friends and starting a business that offered an Uber-like app service that tracked food trucks in and around his hometown in Arizona. And while we did not discuss to the same extent these small ripples with the Fall 2017 focus groups, Black adult learners did want the program to work and at the very least finish something when they had given up on so many opportunities before. This sampling of plans from the spring class focus group demonstrates great desire for independence from whiteness, and tech could possibly lead to that. Or, as DeAndre put simply at the end: "Don't work for anybody else … I like to be in control."

The participants in this ethnographic study understand the work that must be done to topple institutional racism and believe firmly that computer

programming as a new prestigious and emerging literacy will not bring the necessary machinations to make that happen. As they observe from Clearwater Academy's curriculum and assessment, the project of democratizing coding literacy itself participates in and reproduces the structures of racial organizations. The material, financial, and social consequences of how resources in the computer code bootcamp are distributed along specific benchmarks reflective of whiteness in tech leaves the ability to make real change in tech precarious and uncertain. They can be coded as Black functions that meet diversity, equity, and inclusion in the dataset of who works as software developers, but the structures, logics, and practices of tech remain in place. To promote social mobility and this bare minimum node to diversity and anti-racism, Clearwater Academy graduates aren't positioned to rewrite coding literacy for themselves and change how technologies are designed or how relationships form among colleagues and between colleagues and the organizations they work for.

However, visions of what coding literacy may do for themselves, and their communities, aligns with the legacy of sustainability that's endemic in Black rhetorical and literary tradition. They offer a different logic and possibility, one that centers Black tech companies or Black-owned companies who rely on a team of software developers and other folx adjacent to technologies that make their services and products run. Creating resources around Black money and tech also means creating new models for how computer programming and software design can revise racial organizations. This exemplifies transformative access (Banks, 2006), the idea that access to digital technologies is not only about inclusion but also reallocating resources of computer programming to help Black people rewrite the rules of racial systems to the benefit of all people.

Conclusion

In this chapter I've explored the relationship between racial organizations and how they work together to maintain, and possibly transform, societal racial order. The ready-to-work model in coding literacy creates a path toward diversity in tech while reproducing racial hierarchies. Tracking the flow of material and social literacy resources within and between organizations reveal exactly how these systems work. What I have described in this chapter are Clearwater instructors' and Black adult learners' perceptions on what the computer code bootcamp's partnership with tech sponsors means and what the desired outcome may be. While they appreciate the effort to diversify the tech industry, Black adult learners question how they do or do not participate in the outcome tech companies' desire for diverse coders and who really

benefits; some suspect that they need to be shaped into suitable Black people for white employers: people who can speak well and code well. The curriculum explicitly addresses race in only how it relates to getting a job. Instructors sense they can fulfill the mandate to end poverty (one goal of the nonprofit Social Justice Cooperative) but at the expense of leaving the goal to end racism untouched.

I argue that coding Black functions is a fitting metaphor to describe the kinds of coding literacy practice this computer code bootcamp teaches—coding Black function. It suggests that when advocates in the coding movement racialize the call for more opportunities for learning computer programming among historically excluded communities, they must ask what they mean when an institution with all the resources of coding takes responsibility for training these communities. In what ways do existing institutions, upholding certain views on race, recreate Black coders into systems of whiteness? Even an emphasis on the logics of coding through strict focus on web design itself can help maintain racial organizations and implicate Black coders in leveraging their coding literacies for these goals and outcomes. Coding Black functions calls for caution and the need to consider alternative logics rooted in Black lived experiences and knowledge that rewrite the possibilities of what racial schema of computer programming resources can look like. Black adult learners' discourse about their participation in racial organizational power suggests what those alternatives are: in their view, a Black tech ecosystem surrounding and flowing through coding literacy and vice versa would highlight their specific needs, not the needs of tech, and training in computer programming and its many possible career trajectories would rely on Black tech funding. Both would disrupt leaky pipelines from computer code bootcamps to work and diversify the many ways Black adult learners might code themselves.