

## **Designing for assets of diverse students enrolled in a freshman-level computer science for all course**

**Dr. Vanessa Svihla, University of New Mexico**

Dr. Vanessa Svihla is a learning scientist and assistant professor at the University of New Mexico in the Organization, Information & Learning Sciences program, and in the Chemical & Biological Engineering Department. She served as Co-PI on an NSF RET Grant and a USDA NIFA grant, and is currently co-PI on three NSF-funded projects in engineering and computer science education, including a Revolutionizing Engineering Departments project. She was selected as a National Academy of Education / Spencer Postdoctoral Fellow. Dr. Svihla studies learning in authentic, real world conditions; this includes a two-strand research program focused on (1) authentic assessment, often aided by interactive technology, and (2) design learning, in which she studies engineers designing devices, scientists designing investigations, teachers designing learning experiences and students designing to learn.

**Dr. Woong Lim, University of New Mexico**

**Ms. Elizabeth Ellen Esterly, University of New Mexico**

**Irene A Lee, MIT**

**Prof. Melanie E Moses, Department of Computer Science, University of New Mexico**

**Paige Prescott, University of New Mexico**

Paige Prescott has been a classroom science teacher, a curriculum designer and is currently a PhD student at the University of New Mexico in the Organization, Information and Learning Sciences department where she is interested in design experiences for both adults and students as they relate to learning computer science and computational thinking. She regularly conducts teacher professional development for teachers new to computer science and has helped to develop online supports for their continued professional growth.

**Tryphenia B. Peele-Eady Ph.D., University of New Mexico**

Tryphenia Peele-Eady is Associate Professor in Department of Language, Literacy, and Sociocultural Studies, in the College of Education at the University of New Mexico, where she specializes in African American Education and Qualitative and Ethnographic Research.

# Designing for assets of diverse students enrolled in a freshman-level “Computer Science for All” course

## Abstract

Proficiency in computer science skills is crucial for today’s students to succeed in science, technology, engineering and mathematics (STEM) fields and the modern workforce. Despite this fact, few universities count computer science (CS) classes toward the core curriculum. Our university, a Hispanic- and minority-serving research-intensive university located in the American Southwest, recently began counting CS towards fulfilling the laboratory science requirement in the undergraduate core curriculum. This allowed us to consider the characteristics of the students who enrolled in a freshman-level CS course (N=31 students) to identify assets they bring from their diverse life experiences that we might build upon in teaching them. We sought student perceptions of existing curricular modules, in terms of ownership and creativity.

Students completed pre-course surveys about their CS interests, beliefs, prior knowledge and experiences, along with demographics. They completed a brief survey to evaluate some of the modules. We examined descriptive statistics, then conducted tests of difference to identify students’ assets. We explored contrasts between 1) first-generation college students and their traditional peers; and 2) students from historically underrepresented and well-represented groups in computer science. Students who were first in their family to attend college were significantly likelier to agree that CS is important for everyone to study, but also likelier to acknowledge being nervous. This finding suggests that creating a supportive learning environment that enables students to experience relevant CS is integral to retaining first-generation college students in CS. Students from underrepresented groups were significantly likelier to agree that CS is important for solving science problems and for helping people understand problem solving using technology. This finding suggests that our approach, which combines programming and modeling to solve science problems, may be a particularly productive fit for these students.

## Introduction and research purpose

Proficiency in computer science skills is crucial for today’s students to succeed in STEM fields and the modern workforce. Despite this, few universities count computer science (CS) classes toward the core curriculum. Recently, our university, a Hispanic- and minority-serving, research intensive university located in the American Southwest began counting CS towards fulfilling the laboratory science requirement in the undergraduate core curriculum. That our university serves a population overwhelmingly underrepresented in CS provided us with an opportunity to investigate the characteristics and perceptions of students who enroll in a course like this.

## Literature review

We review extant literature on ways to increase diversity in CS, specifically *Computer Science for All* (CS for All) approaches. Through this, we identify a role for asset-based approaches to teaching CS and consider the important potential these approaches have on learning and teaching of STEM disciplines. Likewise, we consider the types of experiences students need to build both initial CS conceptual understanding and creative capacity in CS. We argue these can be understood through the lens of agency, which we define drawing on the social science literature.

## ***Diversity: A resource or barrier for Computer Science for All programs?***

Past efforts, notably Project 2061 [1], that have aimed to educate *all* people (or Americans, students, etc.) in a particular domain (e.g., science, mathematics, etc.), have met with sharp critique due to their colorblind, equality-minded rather than social justice approaches [2]. In response to such critiques, other researchers have argued for the need to consider what would make the domain relevant to each learner [3, 4], that varied life experiences contribute to differences in interest development in school subjects [5], and that learners need opportunities to engage in the disciplinary practices in genuine ways [6, 7]. Others however, have cautioned that such “for all” approaches can be assimilative and reproduce inequities [2, 8]. Bias-blind approaches ignore diversity as a resource and blame learners for not succeeding—a *deficit thinking* approach. Deficit thinking points the finger at the students, rather than critically analyzing structures that prevent students from accessing opportunities to learn [9]. For instance, complaining that students lack sufficient math skills, rather than examining the structures that prevented them from learning, is an example of deficit thinking [10].

As with previous “for all” aims, researchers have raised concerns about the potential for deficit thinking and reproduced inequities in computer science for all projects [11, 12]. Broadening participation in computer science continues to be an area of interest and many efforts have focused on this goal of attracting underrepresented groups to computer science undergraduate courses. For example, some CS educators have successfully increased women’s participation in computer science through inclusive pedagogy in college classrooms [13, 14].

Although there is increasing interest in learning computer science from both students and parents [15-17] barriers to accessing computer science courses in high schools still remain, including lack of course offerings and inadequate technology [12, 15, 16, 18]. When students from groups underrepresented in STEM choose to enroll in an introductory computer science course, they seldom find the topics engaging and relevant to their own lives [18-23]. The computing tasks themselves might not be appropriately leveled, and if students face too much frustration at the beginning of a course, this can negatively impact their self-efficacy in computer science [24], which in turn can impact their persistence in computer science.

However, when faculty consider students’ interests and backgrounds when teaching, students’ enrollment and persistence in computer science courses increase [25, 26]. To accomplish the vision of CS for All, learners need culturally-responsive courses. Such courses emphasize the importance of diverse perspectives in technology development [27]. Further, these courses should begin to develop students’ identities as CS professionals [27], help students connect what they are learning to their everyday experiences, and help them use CS to make changes in areas that matter to them [27]. Such culturally-responsive approaches are effective when teaching CS to African American, Latinx and Native American high school students [28-32].

## ***Asset-based approaches to STEM learning***

In contrast to deficit thinking approaches, asset-based approaches seek to uncover assets students have that can serve as a strong foundation for learning. For instance, consider a student—let’s call her Paloma—who is the first in her family to attend college. Her rural high school had trouble keeping qualified STEM teachers, and her guidance counselor discouraged her from taking any science courses not required for graduation. Despite this, Paloma acquired important

scientific understanding by tending a garden, where she experimented with varying amounts of water, sun, and fertilizer. Paloma's gardening observations could provide her teachers with a foundation on which to shape course content and build formal scientific understanding. Using students' prior experiences as a foundation for further learning not only increases students' interest in CS, but can also enhance the diversity of STEM fields. While relating formal STEM content to students' prior experiences is sometimes challenging, building on students' prior knowledge in STEM practices is typically easier. In this view, students' everyday cultural experiences provide relevant linkages for learning [33].

Engineering has long used asset-based approaches. In one notable example, instructors identified low-income, first-generation community college transfer students' assets to support their matriculation into engineering [34]. Other works reveal how students learned engineering concepts by identifying and solving community needs [35, 36]. In a third example, findings show that by identifying students' assets, instructors supported the success of a diverse group of freshman students in an introductory chemical engineering course [37]; students developed greater engineering self-efficacy and a more accurate perception of engineering practices. Across all of these studies, the assets identified typically included design skills, such as greater facility empathizing with customer, client or stakeholder points of view and more realistic understanding of the role of constraint in design process.

Identifying student assets—rather than focusing on their deficits—paves the way for teaching practices that honor students' backgrounds and cultures. Culturally-responsive teaching recognizes the rich resources students bring with them to class, and focuses on ways to deeply engage these resources, rather than attempting to overwrite them [38-40].

### ***Fostering agency supports learning***

We argue that students need low-agency experiences (i.e., passive watching, limited decision-making) to develop initial confidence and understanding of CS concepts as well as high-agency experiences (i.e., active doing, extensive decision-making) to build creative competence in CS. Like others, we define agency as having opportunities to make and carry out decisions in sociocultural settings [41, 42]. Such settings can be evaluated in terms of their *opportunity structure*—meaning the degree to which learners *are permitted* to make decisions, and the degree to which they actually make decisions [43]. For instance, consider Paloma as she is learning how to program in NetLogo. In her first module, she is permitted to make decisions about the color, size, and location of rosettes that are drawn by the Turtle Graphics program. This approach gives her agency over minor decisions that could lead her to feel pleased by the artistic product; this could be beneficial to her progress, reinforcing her willingness to make and carry out decisions. As such, agency does not equate to free will, but rather to the *recursive process* in which the outcomes of a given action shape future decisions and actions a learner takes [44].

However, we must also consider the scope of the opportunity structure. In many classroom settings, the opportunity structure is present, but very narrow, permitting students to make decisions only about minor aspects of what they are studying. For instance, Paloma, in this first module, was only permitted to make minor decisions. Given that agency is a recursive process, the scope of opportunity structure is very relevant to designing meaningful learning processes. A common approach in educational settings is to consistently provide a narrowed opportunity structure, and sometimes supplement this with non-critical opportunities, such as allowing

students to make decisions about whether to present final work as a poster, pamphlet or presentation.

Because learners cannot effectively direct their own learning in areas that are very new to them, educational settings typically reduce the opportunity structure, effectively lowering expectations for learning. However, this can signal to students that learning in the real world operates this way. With few decisions and limited consequences of those decisions, this can widen gaps or turn students off STEM [45]. Thus, some have argued that classroom settings should include more student-directed learning [46] and open-ended activities [47]. Such activities, while difficult for some teachers to manage [48, 49], provide opportunities for students to think about their futures in ways they might not have otherwise considered [50].

## Methods

We sought to address two research questions: 1) What are the characteristics of the students who enrolled in a freshman-level CS course, and what assets do they bring from their diverse life experiences that we might build upon in teaching them? 2) How do students perceive curricular modules in terms of ownership and creativity?

### *Participants & setting*

Participants included students enrolled in an undergraduate CS for All course (N=31 students) at a research university in the American Southwest. The course counted toward the laboratory science requirement in the undergraduate core curriculum.

### *Course materials*

The CS course was based on a CS for All high school course, which demonstrated that computational modeling is an effective way to teach computer science content and skills while fostering positive attitudes and student engagement in a diverse population (67% of students from historically underrepresented ethnic groups, 36% female students) [51]. Overall, students showed significant increases in learning, and 88% of students agreed or strongly agreed that they liked the course. The CS course teaches programming in conjunction with computational modeling in the programming language NetLogo. Modules develop students' programming skills and introduce computing constructs. Culminating modules require students to build computer models of phenomena then use the model to conduct scientific inquiry. Modules are one week in length with the exception of the final project, which spans several weeks. Several modules are also supplemented with an extension activity. These activities challenge students to apply the knowledge gained in their corresponding module in new and creative ways.

Table 1. Sequence of modules and opportunities for student decision making

#	<i>Module Title</i>	<i>Module description</i>	<i>Student Decision Opportunities</i>
1	Spirograph	Students get acquainted with NetLogo by employing concepts from geometry to generate shapes.	Students choose their own values for angle, step length, and color to generate unique designs.

#	Module Title	Module description	Student Decision Opportunities
2	Hello World	Students write a program that draws a personal logo that includes their name or initials.	Students use any NetLogo commands they choose, including ones they haven't yet learned. They are encouraged to be creative by drawing images that represent themselves and their interests.
3	Spiraling Geometry Using a Repeat Loop	Concepts from Module 1 are revisited; students build on their knowledge by utilizing a repeat loop.	In addition to designing individual rosettes, students choose how many times and in what locations their patterns repeat. Students have agency to employ code reuse and iteration as a tool for open-ended creation in a very simple world.
4	NetLogo Experiments Using Random Walk and Wiggle Walk	Students model movement of contaminants through soil and conduct a series of experiments demonstrating how filtration depends on physical properties of particles.	Students adjust slider bars that they create in order to experiment with how changes in soil properties result in changes in filtration rates.
5	Bumper Turtles	Logic, control and conditional statements are introduced through programming agents. Students program the agents to change their behavior upon encountering a patch of a given color.	Students choose how to design their world through placement and number of differently colored patches to create a unique movement pattern for their agents. In the extension students modify an existing model to create flocks of birds that navigate through obstacles based on their choice of how birds respond to their environment.
6	Saving Nemo: An Ecosystem Model	Students build a model of a simple marine ecosystem and search for ways to keep the system in equilibrium.	Students adjust slider bars that they create in order to experiment with feeding and movement parameters of fish and plankton in order to "save Nemo" by keeping populations in a dynamic equilibrium.
7	Algorithms	Students are given a series of shapes and are tasked with writing algorithms that will follow the outline of each shape.	There are multiple correct solutions to the problems in this module. Students are encouraged to develop algorithms that trace each shape in a variety of ways.
8	Recursion and the Fractal Tree	Recursive processes are introduced as students write code to grow a branching tree.	Students customize their tree structure by controlling branching angles, lengths and widths between parent and offspring branches. Their experiments show small changes have big effects in recursive growth processes.

#	Module Title	Module description	Student Decision Opportunities
9	Spread the Red	Students create an agent-based epidemiological model.	Students determine the characteristics of the virus and hosts they create. They design a model to demonstrate how disease spread depends on the characteristics they choose.
10	Eating Nemo	Students expand Module 6 by adding a big fish species and predator/prey dynamics.	Students choose feeding, movement and lifespans of big and small fish and plankton in order to create an ecosystem that meets a goal that they specify.
11	Project Work: Swarmathon, Networks, or Independent Projects, spanning 3-4 weeks	For independent projects students write their own rubric and craft their own project. Networks project participants define a problem that requires representing networked agents (e.g. rumor spread, food webs) and they simulate how information or resources spread. Swarmathon participants create their own behaviors for robotic agents that compete with classmates' swarms to collect resources.	Students are given complete freedom to define a problem and create a model with which they experiment to solve that problem. Networks and Swarmathon projects walk through several programming exercises that teach necessary building blocks to design networks or robotic agents before they develop their own models. Independent projects often build from prior Ecosystem or Epidemiology modules. Students learn the previous 10 weeks of coding allows them to create arbitrarily interesting and complex models of their choosing.

### ***Data collection & analysis***

Students completed pre-course surveys about their CS interests, beliefs, prior knowledge and experiences, along with demographics. They completed a brief survey to evaluate several of the modules. Survey items were 5-point Likert scale (5 = strongly agree; 1= strongly disagree). We analyzed student work from three modules, attending to the level of detail exemplified in their work, how much they contextualized, and the evidence they used for decision making. We examined descriptive statistics, including the means (*M*) and standard deviations (*SD*). We then selectively conducted tests of difference (independent samples t-tests conducted using SPSS version 23) to identify students' assets. We explored contrasts between 1) first-generation college students and their traditional peers; and 2) students from groups that are historically underrepresented and well-represented in computer science. We also conducted qualitative analysis of student work. We contrasted two modules on current events, focusing on the level of detail/specificity of the modules, both in their programming and written explanations. We explored student agency in the module extensions and final projects, focusing on the opportunity structure of the project options, and the degree to which decisions were consequential (e.g., choosing how to solve a problem) or supplementary (e.g., choosing a color) to the CS content.

### **Results and discussion**

Among our consented students, 43% reported they were the first in their families to attend college. We identified that these students had assets their peers of more traditional backgrounds

were less likely to have. For instance, first-generation college attendees ( $M = 4.46$ ;  $SD = 0.66$ ) agreed or strongly agreed that CS is important for everyone to study, compared to their traditional peers who were neutral or agreed ( $M = 3.56$ ;  $SD = 1.15$ ), and this difference was significant ( $t(29) = 2.55$ ,  $p < .05$ ). Further, first-generation college attendees ( $M = 4.69$ ;  $SD = 0.63$ ) agreed or strongly agreed that creativity is integral to design processes, whereas their traditional peers were neutral or agreed that creativity is integral to design processes ( $M = 3.94$ ;  $SD = 1.06$ ), and this difference was significant ( $t(29) = 2.27$ ,  $p < .05$ ). We also uncovered a difference that could be a barrier to success for first-generation students. First-generation college attendees ( $M = 4.00$ ;  $SD = 1.00$ ) agreed they were nervous about studying CS, whereas their traditional peers were neutral about being nervous, ( $M = 3.22$ ;  $SD = 1.31$ ); this difference was marginally significant ( $t(29) = 1.80$ ,  $p < .10$ ). This measure, in terms of nervousness, suggests that first-generation students may begin with lower self-efficacy in CS. Self-efficacy describes judgments made by students about their ability to be successful in a particular area [52]. Supporting students to develop self-efficacy in an area helps them access more complex information in that area [53, 54]. Essentially, increasing self-efficacy can support students' learning, especially in a trajectory of increasingly agentic tasks.

These findings suggest that creating a supportive learning environment that enables students to experience relevant CS is integral to retaining first-generation college students in CS. Since the students demonstrate a high level of motivation and interest with a relatively low level of confidence, it is important to develop the pedagogical structure in which students can expand their knowledge and skills for CS in the way their initial motivations and interests sustain and transfer to the applications of programming in the real world. The course structure begins with low-agency, simple and well-structured assignments and then builds toward higher agency and more open-ended assignments; this is appropriate for students who are the first in their families to attend college. These students may also benefit from one-on-one support or interactions in the small group setting from peer tutors, teaching assistants, or faculty who could accommodate the knowledge gap in programming skills for those students who are relatively new to programming.

Among our consented students, 53% reported being Latinx, Chicana/o, or Hispanic and 17% American Indian, Alaska Native, or Native American. We identified that these students had different assets than their peers from groups well-represented in CS and engineering. Students from underrepresented groups ( $M = 4.70$ ;  $SD = 0.48$ ) *strongly* agreed that CS is important for solving science problems, whereas their traditional peers agreed ( $M = 4.12$ ;  $SD = 0.78$ ), and this difference was significant, ( $t(25) = 2.12$ ,  $p < .05$ ). Students from underrepresented groups ( $M = 4.63$ ;  $SD = 0.50$ ) *strongly* agreed that helping people understand problem solving with technology is important, whereas their traditional peers agreed ( $M = 4.13$ ;  $SD = 0.72$ ), and this difference was significant, ( $t(25) = 2.17$ ,  $p < .05$ ). These findings suggest our approach, which combines programming and modeling to solve science problems, may be a particularly productive fit for these students. These findings are similar assets others have identified [35, 36]. Further, these findings point to the importance of CS curricula that integrate modeling with problem solving within the local community contexts. With the emerging shift in STEM education from teaching content to engaging students in STEM practices, it is important that university CS programs offer relevant CS curricula that agentially engage students in CS.

We saw evidence of this in student work. For instance, in Module 4, students built models of contaminants percolating through soils. We introduced this Module with a brief description of



the recent Gold King Mine spill on the Animas River. In this close-to-home disaster, over 3 million gallons of gold mine wastewater tainted with heavy metals were accidentally released into the Animas River by the Environmental Protection Agency during routine remediation activities. This water quickly flowed downstream, causing severe contamination that threatened irrigation and drinking water in Colorado and New Mexico. Many students at our university are from communities affected by the Gold King Mine Spill, or know people from these communities. By connecting scientific modeling to current events, particularly direct connections, our students became invested in modeling contaminants in their drinking water, which motivated them to understand how different properties of soils and contaminants would affect the risk of contaminants reaching the water supply in their models. Their models and explanations were more detailed and coherent. In contrast, student work showed less evidence of this when describing the Zika virus outbreak in the Epidemiology Module (9). While both stories were featured in recent news, the local environmental disaster at the Animas River seemed to engage students more than the Zika outbreak, which had no impact on New Mexico. More students agreed that Module 4 was interesting than that Module 9 was interesting, but the difference was not statistically significant. In future work we will include survey questions to more directly gauge how important local relevance is to student engagement and conduct structured observations of in-class engagement.

Our preliminary analysis suggests that building on the interests, experiences, and knowledge that potential CS majors bring with them to class, and connecting curricula to emerging issues can support the learning experiences of students traditionally underrepresented in CS. For example, in the extension of the week 2 module in which students programmed agents to draw their names, students were asked to create a design to reflect something about themselves. Students drew spirals, sine waves and other geometric shapes; some students wrote their names in cursive (one with step-by-step agent instructions, another creating curves from mathematical functions); many drew intricate emblems or logos illustrating aspects of their hobbies, families or academic interests. Students spent more time on the creative designs than the original assignment; this underscores that students are motivated when they have agency toward design they care about. Students' responses to surveys suggest that they learned more when they completed the extensions, specifically about recursion, conditionals and how to program agent interactions with their environments. Students explained the extensions gave them opportunities to be creative.

We also saw evidence of benefits of agentic, creative activity in student work on the later modules, particularly in our analysis of work on the final independent project, which had three options. Two students opted to design their own final independent projects. Interestingly, one of these students, a Hispanic first-generation college attendee, was performing in the middle, academically, and the other, a White male whose parents both attained graduate degrees, was performing at the top academically. These two students both investigated programming techniques and control structures that were new to them. We see this as a reminder that students from a range of backgrounds and abilities may be interested in such work, and that we should not withhold such experiences from students who are lower achieving, but instead provide additional support and feedback to help them achieve their goals.

Ten students opted for the Networks module, which was somewhat prescriptive initially in order to teach students how to create the basic data structures to build, visualize and analyze networks with different topologies (i.e., they built small world networks and measured degree and network

centrality). Students were encouraged to use the ability to build and simulate information spreading over networks to solve a problem of their choice. Students modified their code to model, for example, food webs, social interactions, and family trees. Many students extended or modified a model they created in a previous module. In doing so, they had agency to make choices about which problems to solve and how to solve more complex problems than they had worked on previously. They used building blocks they learned in earlier modules to identify new problems they could solve. For example, one student created a network representing airplane travel and simulated the global spread of disease by incorporating code written for the epidemic module with a spatially situated network model.

Nine students opted to complete the Swarmathon module for their final project, in which they were challenged to design algorithms that controlled a robot swarm performing a foraging task. This module walked students through basic swarm robotic algorithm design, beginning with very simple and prescriptive steps designed to teach basic techniques for deterministic and stochastic search. However, there was no opportunity for creativity during the walk through. While students were encouraged to extend or combine elements of different algorithms presented in the walk through, few students did so, even though such approaches would have provided better solutions. Even the opportunity to compete against the algorithms developed by their classmates did not motivate them to develop new novel algorithms. It appeared to the instructor and course TAs that the design of the Swarmathon module walkthroughs were too simple at the start, and because the problem to be solved was pre-specified, there was no opportunity to develop creative approaches as the students learned basic building blocks. Therefore, students were unprepared to create new algorithms at the end of the 4-week project. In contrast, the Networks module encouraged creativity from the start by requiring students to choose the domain in which they applied the skills they were learning.

### **Significance and implications**

Students from underrepresented groups in CS as well as first-generation college students benefit from studying STEM in a computational modeling format that allows them increasing creativity and agency in defining and solving problems. This accessible approach helps students to invest in their work, which as we argue here, leads to feelings of ownership and belonging. The importance of students having agency in designing their own projects was particularly evident in the final modules. In future work, we plan to investigate ways to provide students with more agentive opportunities in the Swarmathon. Affording students with leeway to define and subsequently solve problems that they find interesting appears to be highly motivating for the population of students in our class.

### **Acknowledgments**

This material is based on work supported by the National Science Foundation (CISE #1240992). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

### **References**

1. American Association for the Advancement of Science. *Project 2061: Science for all Americans*. 1989. AAAS Washington.

2. Barton, A.M. and M.D. Osborne, *Science for All Americans? Science Education Reform and Mexican-Americans*. The High School Journal, 1995. **78**(4): p. 244-252.
3. Fensham, P., *Providing suitable content in the 'science for all' curriculum*, in *Improving science education: The contribution of research*, J. Millar, Editor. 2000, McGraw-Hill Education: UK. p. 147-164.
4. Varma, R., *Making computer science minority-friendly*. Communications of the ACM, 2006. **49**(2): p. 129-134.
5. Sjøberg, S., *Investing all children in 'science for all'*, in *Improving science education: The contribution of research*, J. Millar, Editor. 2000, McGraw-Hill Education: UK. p. 165-186.
6. Jenkins, E., *'Science for all': Time for a paradigm shift*, in *Improving Science Education: The Contribution of Research*, J. Millar, Editor. 2000, McGraw-Hill Education: UK. p. 207-226.
7. Duschl, R., *Making the nature of science explicit*, in *Improving science education: The contribution of research*, J. Millar, Editor. 2000, McGraw-Hill Education: UK. p. 187-206.
8. Aikenhead, G., *Renegotiating the Culture of School Science*, in *Improving Science Education: The Contribution of Research*, J. Millar, Editor. 2000, McGraw-Hill Education: UK.
9. Valencia, R.R., *The Evolution of Deficit Thinking: Educational Thought and Practice*. 1997, New York: Routledge.
10. Davis, J. and D.B. Martin, *Racism, assessment, and instructional practices: Implications for mathematics teachers of African American students*. Journal of Urban Mathematics Education, 2008. **1**(1): p. 10-34.
11. Margolis, J. and J. Goode, *Ten Lessons for Computer Science for All*. ACM Inroads, 2016. **7**(4): p. 52-56.
12. Margolis, J., et al., *Stuck in the shallow end: Education, race, and computing*. 2010: MIT Press.
13. Alvarado, C., Z. Dodds, and R. Libeskind-Hadas, *Increasing women's participation in computing at Harvey Mudd College*. ACM Inroads, 2012. **3**(4): p. 55-64.
14. Redmond, K., S. Evans, and M. Sahami. *A large-scale quantitative study of women in computer science at Stanford University*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.
15. Gallup, *Searching for computer science: Access and barriers in US K-12 education*. 2015.
16. Wang, J., et al. *Gender differences in factors influencing pursuit of computer science and related fields*. in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. 2015. ACM.
17. Wang, J., et al. *Landscape of K-12 computer science education in the US: Perceptions, access, and barriers*. in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 2016. ACM.
18. Goode, J. *Connecting k-16 curriculum & policy: making computer science engaging, accessible, and hospitable for underrepresented students*. in *Proceedings of the 41st ACM technical symposium on Computer science education*. 2010. ACM.

19. Cheryan, S., et al., *Ambient belonging: how stereotypical cues impact gender participation in computer science*. Journal of personality and social psychology, 2009. **97**(6): p. 1045.
20. Grover, S. and R. Pea, *Computational Thinking in K–12 A Review of the State of the Field*. Educational Researcher, 2013. **42**(1): p. 38-43.
21. Margolis, J., et al., *Beyond access: Broadening participation in high school computer science*. ACM Inroads, 2012. **3**(4): p. 72-78.
22. Ryoo, J.J., et al., *Democratizing computer science knowledge: transforming the face of computer science through public high school education*. Learning, Media and Technology, 2013. **38**(2): p. 161-181.
23. Kodaseet, G.G. and R. Varma, *In Pursuit of a Computing Degree: Cultural Implications for American Indians*. Journal of American Indian Education, 2012: p. 67-88.
24. Kinnunen, P. and B. Simon. *CS majors' self-efficacy perceptions in CS1: results in light of social cognitive theory*. in *Proceedings of the seventh international workshop on Computing education research*. 2011. ACM.
25. Dickerson, M., *Multi-agent simulation, netlogo, and the recruitment of computer science majors*. Journal of Computing Sciences in Colleges, 2014. **30**(1): p. 131-139.
26. Barker, L.J., C. McDowell, and K. Kalahar. *Exploring factors that influence computer science introductory course students to persist in the major*. in *ACM SIGCSE Bulletin*. 2009. ACM.
27. Scott, A., et al., *Broadening Participation in Computer Science: Existing Out-of-School Initiatives and a Case Study*. ACM Inroads, 2016. **7**(4): p. 84-90.
28. Ballenger, C., *Social identities, moral narratives, scientific argumentation: Science talk in a bilingual classroom*. Language and Education, 1997. **11**(1): p. 1-14.
29. Meyer, X.S. and B.A. Crawford, *Multicultural Inquiry Toward Demystifying Scientific Culture and Learning Science*. Science Education, 2015. **99**(4): p. 617-637.
30. Bang, M. and D. Medin, *Cultural processes in science education: Supporting the navigation of multiple epistemologies*. Science Education, 2010. **94**(6): p. 1008-1026.
31. Kafai, Y., et al. *Ethnocomputing with electronic textiles: culturally responsive open design to broaden participation in computing in American indian youth and communities*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.
32. Scott, K.A., K.M. Sheridan, and K. Clark, *Culturally responsive computing: a theory revisited*. Learning, Media and Technology, 2015. **40**(4): p. 412-436.
33. Moll, L.C., et al., *Funds of Knowledge for Teaching: Using a Qualitative Approach to Connect Homes and Classrooms*. Theory into Practice, 1992. **31**(2): p. 132-141.
34. Smith, J.M., *Making the Funds of Knowledge of Low Income, First Generation (LIFG) Students Visible and Relevant to Engineering Education*, in *Proceedings of American Society for Engineering Education*. 2015, ASEE: Seattle, WA. p. 1-16.
35. Mejia, J.A., D. Drake, and A. Wilson-Lopez, *Changes in Latino/a Adolescents' Engineering Self-efficacy and Perceptions of Engineering After Addressing Authentic Engineering Design Challenges*, in *Proceedings of American Society for Engineering Education Annual Conference*. 2015, ASEE: Seattle, WA. p. 1-14.
36. Mejia, J.A., et al., *Funds of Knowledge in Hispanic Students' Communities and Households that Enhance Engineering Design Thinking*, in *Proceedings of American*

- Society for Engineering Education Annual Conference*. 2014, ASEE: Indianapolis, IN. p. 1-20.
37. Svihla, V., et al., *Mapping Assets of Diverse Groups for Chemical Engineering Design Problem Framing Ability*. Proceedings of ASEE's 123 Annual Conference & Exposition, 2016: p. 1-25.
  38. Ladson-Billings, G., *Culturally relevant teaching: The key to making multicultural education work*. Research and multicultural education: From the margins to the mainstream, 1992: p. 106-121.
  39. Ladson-Billings, G., *But that's just good teaching! The case for culturally relevant pedagogy*. Theory into Practice, 1995. **34**(3): p. 159-165.
  40. Ladson-Billings, G., *Toward a theory of culturally relevant pedagogy*. American Educational Research Journal, 1995. **32**(3): p. 465.
  41. Ahearn, L.M., *Language and agency*. Annual review of anthropology, 2001: p. 109-137.
  42. Eisenhardt, K.M., *Agency theory: An assessment and review*. Academy of management review, 1989. **14**(1): p. 57-74.
  43. Narayan, D. and P. Petesch, *Agency, opportunity structure, and poverty escapes*. Moving Out of Poverty: Cross-Disciplinary Perspectives on Mobility, 2007: p. 1-44.
  44. Bourdieu, P., *Outline of a Theory of Practice*. Vol. 16. 1977: Cambridge university press.
  45. Olitsky, S., *Structure, agency, and the development of students' identities as learners*. Cultural Studies of Science Education, 2006. **1**(4): p. 745-766.
  46. Martin, J., *Self-regulated learning, social cognitive theory, and agency*. Educational Psychologist, 2004. **39**(2): p. 135-145.
  47. Marcel, A., *The sense of agency: Awareness and ownership of action*. Agency and self-awareness, 2003: p. 48-93.
  48. Svihla, V. and R. Reeve, *Facilitating problem framing in PBL classrooms*. Interdisciplinary Journal of Problem-based Learning, 2016. **10**(2).
  49. Svihla, V., et al., *Framing, Reframing and Teaching: Design Decisions Before, During and Within a Project-based Unit*. International Journal of Designs for Learning, 2016. **7**(1): p. 19-36.
  50. Holland, D., et al., *Identity and agency in cultural worlds*. 1998, Cambridge, MA: Harvard Univ Press.
  51. Lee, I., M. Psaila Dombrowski, and E. Angel. *Preparing STEM Teachers to offer New Mexico Computer Science for All*. in *48th ACM Technical Symposium on Computing Science Education*. 2017. Seattle, Washington, USA: ACM.
  52. Schunk, D.H., *Self-efficacy and academic motivation*. Educational psychologist, 1991. **26**(3-4): p. 207-231.
  53. Svihla, V. and L. Kittinger, *Agentic Trajectories: Development and Learning in a Project-Based High School for Marginalized Students*, in *Transforming Learning, Empowering Learners*, C.-K. Looi, et al., Editors. 2016, International Society of the Learning Sciences.
  54. Boaler, J. *Ability and mathematics: The mindset revolution that is reshaping education*. in *Forum*. 2013. Symposium Journals.