

Learner-Centered and Culturally Relevant Pedagogy

Tia C. Madkins, Jakita O. Thomas, Jessica Solyom, Joanna Goode, and Frieda McAlear

CHAPTER 12

INTRODUCTION: CULTURALLY RELEVANT PEDAGOGY

Underrepresented minority students (for example, Black, Latinx, Native American/Alaskan, Hawaiian / Pacific Islander in the United States) have historically experienced racial bias and structural inequities both inside and outside of school settings. Educational inequities appear at all levels, from low funding for schools with high proportions of underrepresented students of color to diminished teacher and counselor expectations, tracking students into remedial and special needs programs, and over-referring students to school disciplinary officials. For underrepresented students of color, these practices are an extension of colonial and assimilative educational practices, have led to the development of school-perpetuated (historical) trauma, and contribute to experiencing an education environment that feels irrelevant, hostile, and unwelcoming.

Culturally relevant pedagogy (CRP) was first proposed by Ladson-Billings as well as Allen and Boykin in the 1990s. CRP is founded on the idea that learning grounded in a familiar cultural context can potentially increase equitable outcomes. This framework outlines three tenets for academic success: (1) implementing academic rigor, (2) honoring students' cultural and linguistic backgrounds, and (3) helping students to understand, recognize, and critique social inequities. This mode of teaching also emphasizes an authentically caring rapport between teacher and student and connecting curriculum to students' home cultures and everyday lived experiences.

WHY CULTURALLY RELEVANT PEDAGOGICAL PRACTICES MATTER IN COMPUTING

One emerging area of scholarship combines the well-established research and practice of culturally relevant pedagogy with programming education to develop engaging and rigorous programming instruction for underrepresented students of color. This line of research provides a conceptual foundation for integrating culturally relevant pedagogical frameworks into programming instruction across learning contexts. In programming, principles of culturally relevant pedagogy and related approaches include: (1) *supporting student identity development*, (2) *encouraging a critique of inequities in computing*, and (3) *addressing sociopolitical issues*.

Two examples show how this framework can be enacted within the curriculum. In high school, the introductory Exploring Computer Science (ECS) course, designed by Jane Margolis, Joanna Goode, and Gail Chapman, incorporates culturally responsive design tools, such as those explored by Ron Eglash in his work teaching computer science topics to underrepresented groups in culturally responsive ways. ECS offers students ethnocomputing learning experiences. In her account of teaching these lessons, Gayle Nicholls-Ali, a comput-

What we have learned about [swimming and] computer science is that all of the seemingly cultural preferences and interests are profoundly impacted by historical legacies, structural inequities, denied learning opportunities, and belief systems that justify these inequities
– Jane Margolis, *Stuck in the Shallow End*

er science teacher, found that using these culturally relevant educational tools to teach web-based software allowed high school students to apply lessons on algorithms, computing, and how to better use search engines to “create simulations of cultural arts, such as Native American beadwork, as [they] moved from concepts to making, students were excited to finally start ‘programming’ their rugs, baskets, and beadwork [...] with little direction from [the instructor] they deep dived into the website, problem-solved, made mistakes, and iterated.”

In their work in an introductory computer science course with middle school students, Yolanda Rankin and Jakita Owensby Thomas found in their research that integrating a module that leveraged food, recipes, and cooking to expose students to algorithmic thinking as a starting, or anchoring, experience led to 100% retention of Black women undergraduate students for that course.

Such programs suggest engaging underrepresented students of color in educational experiences that strengthen their cultural, linguistic, gender, and racial identities can provide more equitable learning outcomes in computing.

CULTURALLY RELEVANT PEDAGOGY IN PRACTICE

► How We Selected These Practice Examples

Next, we provide practical examples of how teachers have used culturally relevant pedagogy or culturally sustaining practices in programming classrooms in schools, summer programs, or other contexts to support student outcomes. We selected these examples because they exemplify the kinds of culturally relevant or sustaining teaching practices we believe are empowering and important for teaching and learning programming. Importantly, these examples come from our observations of classroom teaching across a variety of contexts that each aim to support students of color in making authentic and meaningful connections between programming and their lives. We desired to provide exemplary instructional examples that educators can use as a guide to support their professional learning and growth. In turn, we hope teachers’ practices can positively influence student learning goals and outcomes (e.g., interest, engagement, achievement) in programming. These examples are sample methods educators can draw and build upon, but they are not prescriptive approaches simply to be replicated. We encourage educators to reflect upon and refine these practices to determine what will work well for their unique schooling contexts. Furthermore, we remind educators that engaging in this work can at times be challenging and messy. Thus, we encourage teachers to integrate these practices into their pedagogical approach with an aim to evaluate and evolve their impact over time.

We organize our ideas and guidelines for CRP in programming classrooms in three categories:

- A. Connect with students’ cultures/life experiences
- B. Empower students to become change agents
- C. Relationships with students, families, and communities

► Connecting With Students’ Cultures/Life Experiences

Rooting computing curriculum and pedagogy in the cultural experiences and the social identities of students allows them to engage and learn about programming in meaningful ways. To integrate students’ rich cultural assets, life experiences, and community knowledge as the building blocks for teaching programming, we have found the following teaching strategies to be effective:

- Engage students with programming activities that are contextualized in students’ school communities (e.g., creating the best transportation route for afterschool activities in their community as a way of learning minimal spanning trees)

- Make programming accessible by connecting learning to students' personal interests, social identities, perspectives, and everyday lives (e.g., inviting a student who skateboards to help introduce a culturally situated design tool about the culture and mathematics of skateboarding)
- Draw from students' cultural assets and knowledge to use as building blocks for examining programming topics (e.g., using popular music to discuss programming paradigms such as loops and linked lists, or using family recipes to discuss how the same algorithm can be represented in different ways)
- Develop project-based assignments for students to both identify an issue of concern to their lives, family, or community and to design a technology-based solution (e.g., mapping food deserts in urban neighborhoods)
- Storytelling from one's own programming education experiences where appropriate to dispel common alternative conceptions of programming concepts and model resilience and effective problem-solving practices in programming education (e.g., storytelling to illustrate the sequence of steps taken to discover that instance variables have a default value in Java)

► Practice Examples: Empower Students to Become Change Agents

Further, providing students with a larger societal view on the impacts and ethics of programming is necessary not only to raise awareness but to also foster student agency to address community problems and inequities that have been created, or might be addressed, by software development. Examples of how this can be done include

- Point out the sociopolitical, not just technical, influences in programming that have led to bias and discrimination being codified in algorithms and software (e.g., face-recognition misgendering rates for people of color)
- Incorporate equity discussions about the impacts of large technology firms on diverse and historically marginalized communities (e.g., what is the impact of Twitter and other tech companies on gentrification and displacement patterns in urban cities like San Francisco?)
- Engage students in critical discussions around data, privacy, surveillance, and other issues with significant impacts on human liberties and civil rights (e.g., police searches of social media accounts)
- Develop students' communication and collaboration skills with programming activities to help students develop and present their ideas and creations to others (e.g., presenting a data science project about an issue in their community using a variety of representations or designing and gathering feedback about a video game designed to address a social issue)
- Discuss and examine the structure of the technological ecosystem to identify potential points of intervention, including the role that education leaders, policymakers, venture capitalists, corporate technology leaders, and start-up founders play in shaping the programming opportunity landscape for underrepresented students of color (e.g., creating a power map of institutions, companies, and organizations that support the entry and success of people of color in tech)

► Practice Examples: Relationships With Students, Families, and Communities

Fostering and maintaining relationships with students and their families is an important aspect of culturally relevant pedagogy. Parents and school communities can play a vital role in supporting and sustaining engagement and participation in programming education for underrepresented students of color.

For indigenous communities, this means that computing experiences involve curriculum that facilitates ongoing interaction and cooperation among key stakeholders, including tribal and community agencies,

families, students, teachers, and school administrators, to promote self-determination for tribal nations. For American Indian communities, computer science learning can assist tribal nations in developing solutions and techno-social agents that can build on the existing infrastructure and address issues unique to their political sovereign status. Given that there are over 570 federally recognized tribes, this variety of geography, cultures, practices, knowledge and thought systems, linguistic practices, and heritage languages, as well as local history, the need for careful contextualization of culturally relevant pedagogies for Native American students and schools is essential.

For educators to engage in relationship-building with students, their families, and their communities, they can:

- Spend time getting to know students and their families outside of the (in)formal classroom setting (e.g., hosting family exposure and recruiting events for programming courses before the fall semester to promote programming curricula, out-of-school programming opportunities, and tech career pathways)
- Recognize that the experiences that students bring with them into the classroom are a form of expertise that should be tapped into and used to ground and help students integrate new content or topics (e.g., discussing students' morning routine as examples of algorithmic thinking)
- Develop project-based assignments for students to both identify an issue of concern to their life, family, or community and design a technology-based solution (e.g., students engage in designing a mobile app that addresses an issue in their neighborhoods, such as job openings for teens in their local area)
- Educators can share personal experiences related to course topics or discussions and/or their trajectories into programming/tech fields (e.g., educator shares experience of being the only woman and person of color in her undergraduate courses)
- Involve families in end-of-year course activities (e.g., invite students to present their final projects to their parents and families at an end-of-year project-demonstration day)
- For schools serving predominantly Native American students, ensuring that culturally relevant curricula and pedagogies are aligned with the goals and self-determination of the relevant tribal leadership and elders is crucial to maintaining engagement with Native American students and their families (e.g., ensuring tribal leadership and elders are provided opportunities to advise and guide computing program designers and staff about culturally relevant computing courses and pedagogical approaches)

CONCLUSION

Culturally relevant pedagogies encourage learning and professional development for teachers with intentions to increase knowledge about and for underrepresented groups, as well as foster a growth mindset among community service providers, teachers, school administrators, and other students. This chapter outlines three major tenets of culturally responsive pedagogy and culturally sustaining practices for teaching computing. More specifically, various practical teaching examples are shared that highlight how computer science and programming concepts and practices can be taught in ways that support student identity in computing, encourage a critique of inequities in computing that youth find important to address, and consider how programming contexts can be used to discuss and address sociopolitical issues in students' local communities. The practices shared earlier—from connecting with students' cultures/life experiences to empowering youth to becoming change agents to building relationships with students, families, and communities to supporting relationships with tribal leadership and communities—all speak to the ways that educators can begin to make computing courses more meaningful and powerful for *all* our youth. Through iterative and self-reflective practices such as those described in this chapter, our computer science teaching community can inspire underrepresented youth to be the creators and innovators, and not simply users, of the technology that can improve our lives and world.

ACKNOWLEDGMENTS

Our gratitude to Yolanda Rankin, Allison Scott, Kimberly Scott, Jean Ryoo, and Alexis Martin for shaping this chapter.

READINGS AND RESEARCH

Building on the foundational work on culturally responsive pedagogy by Allen and Boykin (1992) and Ladson-Billings (1995), educators such as Ron Eglash (2006 and 2013) and Kimberly Scott and her colleagues (2015) have incorporated concepts such as ethnocomputing and intersectionality into their approach to teaching programming to underrepresented students. Goode, Chapman, and Margolis (2012) and Margolis and colleagues (2012) further developed and incorporated learner-centered and inquiry-based teaching practices in the Exploring Computer Science high school curriculum in the United States. For further insight into examples of how programming course content could reflect students' cultural backgrounds and identities, see Rankin and Thomas (2016) and Nicholls-Ali (2017).

BIBLIOGRAPHY

- Allen, B. A., & Boykin, A. W. (1992). African-American children and the educational process: Alleviating cultural discontinuity through prescriptive pedagogy. *School Psychology Review*, 21, 586–596.
- Eglash, R., Bennett, A., O'donnell, C., Jennings, S., & Cintorino, M. (2006). Culturally situated design tools: Ethnocomputing from field site to classroom. *American Anthropologist*, 108(2), 347–362.
- Eglash, R., Gilbert, J. E., Taylor, V., & Geier, S. R. (2013). Culturally responsive computing in urban, after-school contexts: Two approaches. *Urban Education*, 48, 629–656. DOI: 10.1177/0042085913499211
- Goode, J., Chapman, G., & Margolis, J. (2012). Beyond curriculum: the exploring computer science program. *ACM Inroads*, 3(2), 47–53.
- Ladson-Billings, G. (1995). Toward a theory of culturally relevant pedagogy. *American Educational Research Journal*, 32, 465–491. DOI: 10.3102/00028312032003465
- Margolis, J., Ryoo, J. J., Sandoval, C. D., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, 3(4), 72–78.
- Medin, D. L., & Bang, M. (2014). *Who's asking? Native science, western science, and science education*. MIT Press.
- National Congress of American Indians (2020). <http://www.ncai.org/>
- Nicholls-Ali, G. (2017). Baskets, beads, and rugs in computer science. Blog post on *CSForAll Teachers*. Retrieved from <https://www.csforallteachers.org/blog/baskets-beads-and-rugs-computer-science>.
- Rankin, Y. A., & Thomas, J. O. (2016). Leveraging food to achieve 100% retention in an intro CS course. *Journal of Computing Sciences in Colleges*, 32, 127–133. Retrieved from: <https://dl.acm.org/doi/abs/10.5555/3015063.3015083>
- Scott, K. A., Sheridan, K. M., & Clark, K. (2015). Culturally responsive computing: A theory revisited. *Learning, Media and Technology*, 40, 412–436. DOI: 10.1080/17439884.2014.924966