Coding with a Critical Lens: A Developing Computer Programming Curriculum for Diversity and Equity

Bill Kules, University of Maryland, College Park

ABSTRACT

LIS and computer science programs need to address issues of diversity and equity in technical courses like computer programming. This is important because as students transition to their professional careers they will need to understand, navigate, overcome and undo inequitable practices and cultures within their work environment. This paper describes a curriculum to help students recognize, analyze and take action when they encounter these issues. It describes the rationale, framework and structure of the materials, and identifies current challenges. It closes by arguing for stronger, more explicit connections between technical skills courses and program goals related to diversity and inclusion.

TOPICS

Curriculum; Education programs/schools; Pedagogy; Students; Social justice

INTRODUCTION

As LIS and computer science programs expand to educate students for the ever-growing array of jobs in the information professions, they are beginning to address issues of diversity and equity in their computer programming courses. To date, the focus is primarily on how to help students learn programming skills more successfully with course material that is more relevant to the interests of diverse students and by adopting more inclusive teaching practices (Alvarado, Dodds & Libeskind-Hadas, 2012). A few programming courses directly address these issues as part of the course content (Kules, 2017a; Lewis, 2017; Salo, 2016). This is important because as students transition to their professional careers they will need to understand, navigate, overcome and undo inequitable practices and cultures within their work environment (Reynolds & Hartman, 2014).

This paper describes a developing curriculum to help students recognize, analyze and take action when they encounter these issues. It has been used at both the graduate and undergraduate level. This paper describes the rationale, conceptual frameworks, and some practical consideration. It concludes by identifying some of the challenges and arguing for stronger, more explicit connections between technical skills courses and program-level diversity and inclusion themes.

RATIONALE

There are compelling ethical and practical reasons why information professionals need to understand these issues in their organizations and communities (Forsgren & Humble, 2016; Sinclair, 2004; Wajcman, 2009; Wolske, Rhinesmith & Kumar, 2014). Within organizations, the value of diverse teams is well established (Phillips, 2014), but organizational success depends on teams managing diversity effectively (Jackson & Ruderman, 1995). Programmers and other technical professionals will be more effective team contributors if they understand how these issues intersect with team dynamics. Thus an important element of this curriculum is helping students to understand the dynamics of teams and particularly the relationship to issues of team culture and individual bias.

FRAMEWORKS

The curriculum uses two primary conceptual frameworks: social justice teaching and organizational/team dynamics. The social justice approach addresses issues of social identity and how this impacts power relationships and confers advantages or disadvantages. It helps students to recognize and analyze issues more deeply than common approaches to diversity, which emphasize cultural and social differences and commonalities (e.g., cultural competency) without necessarily addressing issues of inequality (Adams & Zúñiga, 2016). Structural inequality occurs at multiple levels – individual, institutional, cultural (Hardiman, Jackson, & Griffin, 2013) and reinforces unearned, inequitable, and often-unrecognized forms of privilege and oppression (McIntosh, 1988).

All of these elements are evident in teams. Teams reflect their organization, but team culture is more easily changed than the larger organizational culture, so they provide a useful entry point for this curriculum. We already use small groups extensively in our coursework so they provide a natural learning environment where patterns of privilege and oppression emerge. By analyzing and acting upon these issues within their groups, students can develop skills in a supportive environment, where mistakes are recognized as learning opportunities. Connecting understanding to strategies for action provides a way for students to feel empowered to take action.

STRUCTURE

The initial curriculum was part of a graduate level introduction to JavaScript course taught in Spring 2016. It has been refined and used in five more classes, including two semesters of a mid-level undergraduate Python course and one section of an introductory undergraduate Python course. The learning outcomes capture two essential elements of knowledge and skill:

1. Explain how programming is situated in and reflects broader social structures, constructs and issues, e.g. race, class or gender.
2. Within their teams and small groups, notice when inequities surface and take positive action to work with their peers to resolve them.

Readings and activities are used for weekly reflective discussion on the "bigger picture" of computer programming. We introduce a reflective practice at the beginning of the semester, starting with more pragmatic questions focused on the programming language and computational thinking concepts. The diversity and equity elements are introduced about half way through the semester, after the students have settled in and gotten to know each other. At the end of the semester students write a final essay analyzing one example of a diversity or equity issue in technology.

Discussion topics include:

- Coding for social good
- Coding in its social context
- Systems of power in tech: individual, organization, culture
- Forms of inequity, unearned privilege and oppression in tech
- Taking action and forms of resistance in tech
- Team dynamics - structures to support equitable practices

We draw readings from a variety of sources. We avoid scholarly journal articles in favor of shorter, more engaging formats such as blog posts, opinion pieces and popular press articles. Samples of readings and discussion prompts include:

- *How to Hold Governments Accountable for the Algorithms They Use* (Diakopoulos, 2016) – Algorithms determine prison sentences and Social Security benefits. So we need to know how they work. What does this tell us about the power and use of algorithms?
- *Missed Connections: What Search Engines Say about Women* (Noble, 2012) – Algorithms can reinforce existing social and cultural bias. How do we respond as programmers and technology designers?
- *How Diversity Makes Us Smarter* (Phillips, 2014) – Research shows that socially diverse groups (that is, those with a diversity of race, ethnicity, gender and sexual orientation) are more innovative than homogeneous groups. Diverse teams also present challenges. Why is this so? How have you personally experienced diverse groups?
- *Google's Ideological Echo Chamber* (https://assets.documentcloud.org/documents/3914586/Googles-Ideological-Echo-Chamber.pdf) and *The e-mail Larry Page should have written to James Damore* (The Economist, 2017) – The memo and this response illustrate the ongoing debate about what diversity means in tech companies generally. Damore's memo describes his experience at Google and his critique of diversity efforts there. He was subsequently fired, leading to a public dialog. The Economist published a point-by-point rebuttal to his arguments. How compelling are Damore's and The Economist's arguments for you? Do you agree with their conclusions? Disagree? Why?
- *What If I Had James Damore of Google on a Team?* (http://www.incontextdesign.com/what-if-i-had-james-delmore-of-google-on-a-team/) – This post reflects on the challenges of working in diverse technology teams and presents six techniques to help diverse teams work together. As you work on your team projects, have you noticed any of these issues? Consider how you can incorporate these techniques into your project team.

DISCUSSION AND CONCLUSION

The discussions with students are tremendously satisfying, but there are a number of challenges in teaching this curriculum. It requires changing the way the course is taught by using more inclusive pedagogy (Kules, 2017b; Alvarado, Dodds & Libeskind-Hadas, 2012). It takes time to develop a level of trust within the class, and not all students are willing to engage. Student essays and course evaluations reflect a range of reactions and levels of growth. Some students are enthusiastic and grateful for the opportunity to discuss programming in a larger context. They find it meaningful and motivating. Some students continue to question the rationale. One current challenge is to help students – especially more privileged students – recognize how this is relevant to their own careers. Overall, most student essays demonstrate an ability to recognize and analyze diversity and equity issues.

Developing this curriculum has stimulated conversations within the iSchool. Presentations have prompted faculty colleagues to discuss how diversity and equity themes could be integrated into their own courses. It has also provided an opportunity for discussions with the student diversity group, iDiversity. In turn, this led to significant contributions to the readings and suggestions on structure.

As the number of students in our undergraduate program continues to grow, we are offering additional sections of these courses, which are being taught by more instructors. Instructors have varying levels of knowledge, skill and comfort with the issues and the teaching approaches needed to do this work. It requires techniques that differ from those used in a traditional skills-focused course. Instructors who have primarily taught programming as a skills course are challenged to expand their teaching approach, and not everyone is prepared to teach this material. It can be emotionally charged and personally unsettling, as well as transformative (Bell, Goodman, & Ouellett, 2016). This highlights the need for professional development for faculty – and for instructors to commit to "doing our own work." As instructors, this entails understanding our own individual social identities, experiences of privilege and oppression, and how this impacts not only our teaching, but also our own experiences as programmers or creators of technology so that we can authentically and effectively engage with the material and our students (Bell, Goodman & Varghese, 2016).

In the larger academic context, this curriculum demonstrates one way for programs to respond to the ongoing challenge in LIS education of meaningfully engaging curricula with issues of diversity, inclusion and equity (Jaeger et al., 2015). This can be visualized as a part of a "T-shaped" curriculum. Courses focused on diversity and equity can provide deep understandings, while other courses, like this one, examine how they intersect with the course topic. This can help students develop the technical and ethical skills needed to be successful as they move into their careers as information professionals.

ACKNOWLEDGMENTS

Elizabeth Lieutenant, Courtney Douglass and Coral Sheldon-Hess made generous contributions to this curriculum. Kanika Mehta provided additional research assistance.

REFERENCES

Adams, M., & Zúñiga, X. (2016). Getting started: Core concepts for social justice education. In Adams, M. & Bell, L.A. (Eds.), *Teaching for diversity and social justice* (95-130). New York: Routledge.

Alvarado, C., Dodds, Z., & Libeskind-Hadas, R. (2012). Increasing women's participation in computing at Harvey Mudd College. *ACM Inroads* 3, 4 (December 2012), 55-64.

Bell, L.A., Goodman, D.J. & Ouellett, M.L. (2016). Design and facilitation. In Adams, M. & Bell, L.A. (Eds.), *Teaching for diversity and social justice* (55-93). New York: Routledge.

Bell, L.A., Goodman, D.J. &, Varghese, R. (2016). Critical self-knowledge for social justice educators. In Adams, M. & Bell, L.A. (Eds.), *Teaching for diversity and social justice* (397-418). New York: Routledge.

Diakopoulos, N. (2016, February). How to Hold Governments Accountable for the Algorithms They Use. *Slate.* Retrieved from http://www.slate.com/articles/technology/future_tense/2016/02/how_to_hold_governments_accountable_for_their_algorithms.html

Forsgren, N. & Humble, J. (2016) The Core Belief Keeping Marginalized Groups Out of Tech. *Model View Culture 32.* Retrieved from https://modelviewculture.com/pieces/the-core-belief-keeping-marginalized-groups-out-of-tech

Hardiman, R., Jackson, B., & Griffin, P. (2013). Conceptual foundations. In M. Adams, W.J. Blumenfeld, C. Castañeda, H.W. Hackman, M.L. Peters, & X. Zúñiga (Eds.), *Readings for diversity and social justice* (3rd ed., 26-35). New York: Routledge.

Kules, B. (2017a). Teaching JavaScript as social justice: Interrogating culture, bias and equity in an introductory programming course. Paper presented at the Annual Conference of the Association for Library and Information Science Educators (ALISE 2017) (Atlanta, GA, January 17-20, 2017).

Kules, B. (2017b). Critiquing culture, bias and equity in introductory computer programming courses. Presentation at the Lilly International Conference on Evidence-Based Teaching and Learning (Bethesda, MD, June 1-4, 2017).

Jackson, S. E., & Ruderman, M. N. (1995). *Diversity in work teams: Research paradigms for a changing workplace*. American Psychological Association.

Jaeger, P. T., Cooke, N. A., Feltis, C., Hamiel, M., Jardine, F., & Shilton, K. (2015). The virtuous circle revisited: injecting diversity, inclusion, rights, justice, and equity into LIS from education to advocacy. *The Library*, *85*(2).

Lewis C.M. (2017) Good (and bad) reasons to teach all students computer science. In: Fee S., Holland-Minkley A., Lombardi T. (eds) *New Directions for Computing Education*. Springer, Cham.

McIntosh, P. (1988). *White privilege and male privilege: A personal account of coming to see correspondences through work in women's studies*. Wellesley, MA: Wellesley College, Center for Research on Women.

Noble, S. (2012, Spring). Missed Connections: What Search Engines Say about Women. *Bitch magazine*, *12*(4), Spring, 37-41. Retrieved from https://safiyaunoble.files.wordpress.com/2012/03/54_search_engines.pdf

Phillips, K. W. (2014). How diversity works. *Scientific American*, *311*(4), 42-47. Retrieved from https://www.scientificamerican.com/article/how-diversity-makes-us-smarter/

Reynolds, R. H. (Director/Producer) & Hartman, S. (Producer). (2014). *Code: Debugging the Gender Gap.* [Motion picture]. (http://www.codedocumentary.com/)

Salo, D. (2016). *LIS 640: Code and Power* [syllabus]. Retrieved from http://dsalo.info/pdfs/uploads/2016/01/640CodePowersyll2015.pdf.

Sinclair, B. (2004). Introduction: Integrating the Histories of Race and Technology. Technology and the African-American Experience: Needs and Opportunities for Study. Cambridge: MIT Press. Retrieved from https://mitpress.mit.edu/sites/default/files/titles/content/9780262693448_sch_0001.pdf.

The Economist. (2017, August 19). *The e-mail Larry Page should have written to James Damore* [Editorial]. Retrieved from https://www.economist.com/news/international/21726276-last-week-newspaper-said-alphabets-boss-should-write-detailed-ringing-rebuttal

Wajcman, J. (2009). Feminist theories of technology. Cambridge Journal Of Economics, 34(1), 143-152. doi:10.1093/cje/ben057

Wolske, M, Rhinesmith, C. & Kumar, B. (2014). Community Informatics Studio: Designing Experiential Learning to Support Teaching, Research, and Practice. *Journal of Education for Library and Information Science 55*(2). 166-177.