Teaching JavaScript as Social Justice: Interrogating Culture, Bias and Equity in an Introductory Programming Course

Bill Kules

University of Maryland, College Park

Keywords

Computer programming; social justice; education; critical reflection

Short abstract

When learning skills like computer programming, students need to develop an understanding of issues of culture, bias and equity at the same time that they learn the technical elements. As information professionals they will need to understand and navigate these issues. This presentation describes a course that integrates both social justice and technical elements, instead of separating them into different courses as is typical practice. I will describe the approach and structure and reflect on the experience and student feedback.

Abstract

There are no curriculum boundaries when addressing issues of race, class, and gender.

–Brenda Collins Flyswithhawks (1996)

Our students need to develop an understanding of issues of culture, bias and equity at the same time that they learn technical skills, like programming. There are compelling ethical and practical reasons why information professionals have a responsibility to understand the organizations and communities in which we use programming skills (Forsgren & Humble, 2016; Sinclair, 2004; Wajcman, 2009; Wolske, Rhinesmith & Kumar, 2014). We don't teach reference services, project management or cataloging isolated from their organizational and culture contexts, and we should similarly be teaching programming in its cultural context. Unfortunately there are few examples of courses that integrate these themes in a meaningful way. This reflects an ongoing challenge in LIS education to meaningfully engage curricula with issues of diversity, inclusion and equity (Jaeger et al., 2015).

People of color, women, and other marginalized groups are underrepresented in programming and related fields, and are systematically pushed out of and away from programming, starting at least as early as middle school, and continuing through high school, college and all the way to senior level positions (Reynolds & Hartman, 2014; Margolis, Estrella, Goode, Holme & Nao, 2010; Margolis & Fischer, 2003). The obstacles they navigate range from individual bias and harassment (e.g., Gamergate) to structural (e.g., lack of computer science courses in lower-income schools). It is important for all of us to understand and work to dismantle these obstacles.

There is plenty of inspirational work underway outside of academia – with initiatives like the Techies Project (http://www.techiesproject.com/), Black Girls Code (http://www.blackgirlscode.com/), All Star Code (http://www.allstarcode.org/) and PyLadies (http://www.pyladies.com/), as well as professional organizations like the Association for Computing Machinery (ACM) providing networking, support, mentoring, hackathons, and conferences that highlight and challenge racism and sexism in the programming field.

Introductory computer science courses within a few innovative programs are being transformed in how they address pedagogy. They are adopting computational approaches instead of solely focusing on learning a specific programming language, to "provide students with substantial programming experience in a variety of application areas and exposure to some of the major intellectual and societal contributions of our field" (Alvarado, Dodds & Libeskind-Hadas, 2012). Additionally, they are recruiting students with trips to the Grace Hopper Celebration of Women in Computing and providing early research experiences for women.

What's missing – at least for Master's students who will soon be joining (or re-joining, or are already immersed in) professional contexts, where the cultures reflect much of the negative status quo – is information these professionals will need to critique, negotiate, navigate and change these cultures. Practical programming skills are generally separated from social justice or ethical issues, often in separate departments or schools. There are few examples where the social and culture context is interrogated in conjunction with learning programming. Critical code studies (Marino, 2006), for example, interrogates code and technological systems, but doesn't teach programming. One exception is a University of Wisconsin-Madison course, Code and Power which teaches PHP programming within a larger framework that critiques "economic, social and cultural structural mechanisms related to racial and gender disparities in the computing industries" (Salo, 2016).

Incorporating issues of culture, bias and equity into a graduate level introductory JavaScript programming course entailed changing both *what* and *how* I teach. My objective is to provide an inclusive, welcoming and supportive environment for learning, especially for students new to programming, to counteract a lifetime of misinformation about the difficulty of programming and who can program. In this presentation I will describe the questions I asked as I prepared and the resources I drew on to structure the course. I will reflect on my experience and student feedback, and how my own social identity (white, male, middle class, etc.) affects the assumptions I make about what, who and why I teach. I will invite the audience to reflect on the opportunities and challenges for integrating these issues into our technical courses as we seek to provide supportive and challenging environments to learn the knowledge and skills needed to be well-rounded, critically aware professionals.

References

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.

Flyswithhawks, B. C. (1996). The Process of Knowing and Learning: An Academic and Cultural Awakening. *Holistic Education Review*, *9*(4), 35-39.

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

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).

Marino, M. C. (2006) Critical Code Studies, Electronic Book Review, Retrieved from http://www.electronicbookreview.com/thread/electropoetics/codology

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2010). *Stuck in the shallow end: Education, race, and computing*. MIT Press.

Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. MIT press.

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.

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.