# DIVERSITY STATEMENT

Kristopher Micinski (kris@cs.haverford.edu)

Computing-related majors and careers have long-standing issues with diversity. Beyond recognizing diversity as a problem for computer science as a whole, it also deeply matters to me personally. Especially as I do not belong to a diversity group, I continually educate myself by reading books and blog articles on the subject and having frequent conversations with my underrepresented colleagues. Below are some of my experiences attempting to foster inclusivity.

Diversity comes in many forms, and I have been careful to avoid assumptions about student identity based on minority status before more closely understanding their situation. Even students who are apparently in the majority may have issues such as learning disabilities that make them part of a diverse group. At the beginning of my courses, I reach out to all students to assess how their identity might affect the course for them. I provide the opportunity for anonymous feedback and ensure students know they can voice concerns without retribution.

Students may have different needs based on their backgrounds. For example, not every student will come from come from a background that supports their computing education before they enter university education. One way I have assuaged this concern is to make classes self contained to the extent possible. For example, in giving explanations to students during class I have avoided drawing upon material—such as references to programming languages or frameworks—that has not been covered in prerequisite courses but which may be familiar to students privileged with a background in computing or who do it as a hobby. My colleagues and students have told me these references made them feel that they were implicitly behind other students. I emphasize that a background in computing is not necessary to be successful, and actively point out role models who lacked this background.

Encouraging inclusivity in classes requires an environment where students feel they will be heard. Rather than soliciting answers from any student in class—which literature shows discourages women and minority participation—I break students into groups to discuss questions and sample groups to answer. This strategy has worked: instead of only majority students (frequently men who sat in the front of class) answering questions, students who had been previously more reserved (frequently women and minorities) began to participate as well.

I have actively worked with underrepresented students to ensure course logistics do not hinder their participation. I respond quickly to answers via forums such as Piazza. I also make myself available for office hours via Skype for students that might not be able to attend regularly. One issue at Haverford is that students from Bryn Mawr (a womens college a few miles away) feel the course is less accessible. To assuage this, I offer a variety of office hours at coffee shops in Bryn Mawr.

Along with changes to course structure and content, working directly with minority students increases their confidence and feeling of belonging in the field. I have mentored multiple students students as part of REU programs and summer research for high school students. These programs offered me a great opportunity to mentor students more closely than during the term. The students I worked with have said the experience has made them feel more confident in their abilities and more comfortable approaching faculty for help. One student even said the experience gave them the confidence to pursue a STEM major when they had previously been uncertain. This student successfully graduated and is now a successful software engineer.

Last, I realize that biases are often established long before students select university majors. I currently volunteer at CodedByKids in Philadelphia, an outreach program for youth in the city. CodedByKids meets at computer labs in community centers (accessible to everyone) and I work one-on-one with students to teach programming skills such as web development and JavaScript programming.