1 Teaching and Learning Philosophy

I was fortunate to begin my teaching career earlier than most when, as an award-winning teaching assistant, my department gave me the opportunity to teach entire courses while I was still a Ph.D. student. From day one, I found myself passionate about the content and taught with great enthusiasm. However, with time, I came to realize that my teaching practices were arbitrary and lacking evidence of effectiveness, except for strong USRI scores. Just because I enjoyed the material and was enthusiastic about teaching and learning, these characteristics did not necessarily transfer to my students. I was realizing that while students were one of many variables in the teaching-learning equation, they were the most important variable. Over the next few years, I became convinced that my teaching must transform to become more student-centric, fostering a deep learning approach [Marton and Saljo, 1976], and thriving to always produce a lasting effect on the learners' experience [Ramsden, 2003].

In a deep approach to learning [Marton and Saljo, 1976], learners tend to be intrinsically motivated; they focus on the big picture, analyzing and evaluating alternatives. On the contrary, learners who adopt a surface approach memorize facts without reflection and prepare themselves to artificially succeed in exams. My current teaching philosophy has evolved around how to create a fertile environment in which students learn deeply. My teaching practices are now founded in four concepts: (1) engaging students, (2) fostering collaborative learning, (3) valuing feedback, and (4) encouraging self-directed learning.

1.1 Engaging students

I believe that in order to promote a deep learning approach, learning should be an engaging process for both students and educators. For students, active engagement in the learning process allows them to develop deeper understandings of concepts and ideally of themselves as learners. I engage my students by adding the "fun factor" to the process, without sacrificing the "challenge factor." I began using class-response systems (such as eInstruction and Top Hat) in 2009 for quizzing students and polling them on various issues relevant to their learning experiences. While quizzes give me immediate feedback on my students' understanding of lecture topics, they also create an engaging class environment for students. Furthermore in a recent best paper award winner SoTL study, we demonstrated how such class-response systems can be used to help students retain course content [Collier and Kawash, 2017].

Research suggests a strong correlation between the ability to solving game-like puzzles and real-life problems [Falkner et al., 2010]. I redesigned my CPSC 203 (*Introduction to Problem Solving Using Application Software*) lectures around intertwining puzzle-based and problem-based learning. Beginning my lectures with game-like puzzles that students attempt to solve in groups creates a lively classroom atmosphere with interesting and enthusiastic discussions. I select the puzzles carefully so they relate to the lecture's topic and set the stage for the more advanced topics. Such class activities are used as an alternative to providing content-filled courses. Overloaded course content has been shown to encourage a surface approach to learning [Ramsden, 2003]. Research also suggests that the majority of Computer Science students enjoy learning while developing interactive video games, which incorporate many of the basic principles in Computing Sciences. In CPSC 359 (*Computing Machinery II*), I challenge students to develop interactive video games as the major assignment. While students typically consider the material in CPSC 359 as complex and dry, I have witnessed them embark on their projects with great enthusiasm since I started requiring the game project.

I also engage my students by appropriate use of social power [McCroskey et al., 1985]; use of open-book exams to promote deep learning [Biggs and Tang, 2007]; and employing group exams and projects to foster collaborative learning.

1.2 Fostering collaborative learning

I believe that most students learn deeper in smaller groups. In all of my courses, I either require or encourage students to work in groups. My first class of the semester is an "ice-breaker," where I ask students to work together on a fun problem that is related to the course material. Additionally, I ask my students to work on Top Hat questions in groups, and the puzzles are also done in groups. In CPSC 457 (*Principles of Operating Systems*), I utilize group exams for a limited number of flipped classes, in which students prepare for class by studying a certain subject in detail and then do active learning during class hours. Outside the classroom, I have students work in teams for their project assignments, requiring one submission per team. The teams may change between unrelated assignments so they interact with more of their classmates, experience different learning preferences, and finesse their team skills. Such measures have been shown to foster a deep approach to learning [Biggs and Tang, 2007]. From my own industry experience, team skills are essential for students' success in their future jobs.

1.3 Valuing feedback

I collect real-time student feedback throughout the semester as USRI results are only available when the course is over. In addition to my regular informal chats with students, I use quick Top Hat polls for my immediate (same-class or same-week) decisions. I also employ Top Hat or D2L polls to guide my same-term decisions, and I conduct more thorough (paper) surveys to assess specific class practices. The collected results help me accomplish informed course re-designs and report the results to the broader Computer Science education community as SoTL.

Feedback from different sources is very crucial for student learning and success. My projects are designed so that instructor and TA feedback can be given at various stages of development; peer feedback within teams and intra-teams is abundant; demos for projects help provide real-time feedback to students; and showcasing of projects in different venues provide constructive use of reward power [McCroskey et al., 1985].

1.4 Encouraging self-directed learning

Ultimately, I regard myself as a successful educator when I help my students become self-directed learners. In the dynamically changing Computer Science field, I believe that self-directed learners have better chances for success in the workplace. While many of my project assignments tend to be open-ended, I nonetheless provide a framework with clear objectives and requirements. However, students choose their own topics and make their own decisions about how to approach a certain problem. This requires close supervision on my behalf to keep students on track. While this approach may be easier to implement in senior courses (such as my CPSC 457 assignments), it is possible in introductory courses, as I demonstrated in CPSC 203 and CPSC 359. This strategy encourages students to think independently and explore new territories on their own, providing a fertile environment to further deepen their learning. Often, some students approach me after the semester ends asking for extending their permission to access the lab to that they continue working on their projects so that their projects can be showcased on my YouTube channel.