

Teaching Statement

My goal in classes I teach is for students to be able to appreciate the mathematical foundations of programming languages and their application to real world programs that they write themselves. I have been involved in teaching the following courses at NUS – Programming Language Concepts (CS2104), Programming Language Implementation and Logic (CS4215) and Formal Methods (CS3234). In these courses I have tried to instill the importance of static typing, reasoning about correctness of code and thinking in a functional programming style.

My teaching philosophy in these courses can be best described as “Learning by Programming”. I have designed tutorials and lab sessions for students that introduce key concepts from the lecture in a practical context. The focus of tutorial questions is to help students practice writing programs that illustrate various programming language features and how those features relate to building a solution for the problem of their interest. The homework assignments are designed to test students on their programming skills and feature bonus questions for those who want to practice more. Giving extra credit for bonus questions or rewarding students for doing challenge problems help to keep the class engaged and exciting throughout the semester. In addition to handing tutorial and lab sessions, I have also given guest lectures in two of the courses on the following topics – Introduction to OCaml Programming and Automated Verification with HIP/SLEEK. During lecturing I encourage questions and have stop slides to provide time for discussions and reflection on the topic. My effort is to make the class more vibrant by providing a forum for dialogue between students and teacher. In order to show the significance of the class in the broader scope of the software industry, I always include examples (code snippets, bugs, proofs etc.) from popular open source projects (Linux Kernel) or recent issues about software (HeartBleed OpenSSL bug).

I believe that providing the right kind of tool support can enhance the learning experience for the students. For this I have developed TeachHIP (<http://loris-7.ddns.comp.nus.edu.sg/~project/TeachHIP/>), a web based interface for teaching automated verification to students using HIP/SLEEK verification system. TeachHIP was used this in CS3234 for verifying sample demos during lecture as well as solving tutorial problems. In addition, the students used the system to do their lab assignments as well. The tool was well received by students who wanted access to the tool even after the course to help them verify programs beyond the class. Teaching support tools also make it easier to cater to different speed and learning patterns of students. In order to give individual attention to those students who need it, I also provide time for consultation. The individual consultation times are essential for some students who may otherwise be struggling with the course or stuck on some hard problem. My focus in these consultations is to treat students as equals and help them just as one would help a colleague or coworker on some problem at work. These consultations are also times when students feel less inhibited and ask more frank and honest questions.

My teaching experience has been a valuable and enriching learning for me as well. I have found that true understanding of a concept is only attained when you can explain and teach it to someone else. In future I would also like to try out some of the different styles of teaching (like flipped classroom).