Fun and Frustration – One Year Teaching at HKUST

Fleischer, Rudolf

rudolf@ust.hk
Department of Computer Science,
Hong Kong University of Science and Technology

ABSTRACT

I will talk about some of my experiences in my first year teaching computer science at HKUST. I will describe some of the fun I had trying to adapt new teaching concepts to the different student environment in Hong Kong, and some of the frustrations I experienced in the process.

BACKGROUND

After spending one year at the University of Waterloo in Canada and many years at the University of Saarbrücken Germany, I joined the Department of Computer Science at HKUST in Fall 2000.

'HONG KONG STUDENTS ARE DUMB'

The first thing my new colleagues warned me about was that "Hong Kong students are dumb, compared to students at North American universities". Always the optimist, I did not want to believe this immediately, but my first encounters with Hong Kong students in the classroom gave me second thoughts. Indeed, the students were definitely more passive and unable (or reluctant) to answer even simple questions compared with the students I knew from Canada or Germany. But I soon realized that, at least in some cases, it was communicating in English that was actually their main problem.

ACTIVATING STUDENTS

So in my next course I tried to force the students to become more actively engaged. For the tutorials, I adopted a framework that I knew works pretty well at my old university back in Germany. There the students are not treated to another hour or two of lecturing in the tutorials, this time by the TA instead of the professor (which I believe does not improve the situation), but the students themselves are asked to step forward and present their solutions to the assignment problems at the blackboard (or here at HKUST, at the whiteboard). I attended all tutorial sessions and, after a few weeks, the students had risen to the challenge, some of them even regularly

volunteering to present their solutions, even though they knew perfectly well they were wrong. The ensuing discussions were great fun, and I think that the students benefited from discussing the wrong solutions.

CLASS REPRESENTATIVES

Most Hong Kong students are pretty shy. They rarely dare ask questions, either in the classroom or after class during office hours. So it can be hard to guess how a course is going. Therefore, this term I tried to have two class representatives elected who can act as the mouthpiece of the class, communicating to me all problems with the course. Finding these two class representatives was a really frustrating exercise. As you might expect, nobody volunteered, so it took a lot of arm-twisting to push two students to accept the post. But now that they are established they do a great job (you wonder why they were so reluctant in the first place), and I can react much better to the problems in class than otherwise (because I hear about them early on from my two class representatives).

ACTIVE LEARNING

For the theoretical computer science courses I usually teach, Problem Based Learning (PBL) is not a good option. But on a smaller scale, Active Learning is a good way to engage the students more actively. At certain points in my lectures, when the students believe they have halfway understood what is going on, but actually have not yet really understood it, I stop the lecture and ask the students to discuss a tricky question in small groups (usually the table neighbors). Usually, many (but unfortunately not all) students engage in heated discussions, quickly realizing that things are more difficult than they seemed to be a few minutes earlier.

MARKING ASSIGNMENTS

I believe that written homework assignments are an integral part of the learning process. Therefore I do not like the idea of marking them for correctness (how could we possibly do that if the students have not finished learning the new material?). Instead, students get full marks for showing effort in solving the problems, even if the solution turns out to be incomplete or wrong. Testing knowledge and understanding of the material is left to the exams that are, of course, marked in the usual way.

My experiences in Germany and Canada show that, under this marking regime, the students are much more motivated to work out their own solutions, i.e., it reduces the amount of cheating because there is no pressure to hand in the `right' solution. Clearly, students learn more from their own efforts to solve a problem than from copying a solution. I also encourage the students to work in small teams (two to four students) and hand in team solutions because discussing a problem with others usually helps them to better understand a problem and its solution. And formalizing teamwork

makes the students feel less guilty, because many would work together even if it were not allowed.

To challenge the good students in the class, each assignment also contains a challenge problem for a bonus point. These problems are not necessarily related to the course material, but the students can practice their combinatorial problem-solving skills on them.

SELF- AND PEER-ASSESSMENT

I am currently supervising twelve final-year projects. Each project team (two to four students) has to write three reports over the course of the project. I have asked the students to self-assess their own reports and to peer-assess another group's reports. I provided a grading sheet with detailed questions on the content and quality of the reports. Not surprisingly, in the first round, most teams valued their own work 5% - 10% higher than the mark they got from their fellow students. But in most cases, their marking was fairly objective.

The self-assessment should normally force the students to first of all write a better report, whereas seeing good or bad reports from other groups should help them write a better report next time, copying the good things they saw and avoiding the bad things they noticed. Also, grading a report should improve their critical thinking and their judgment capabilities.

HONG KONG STUDENTS ARE NOT SO DUMB

I put my colleagues' claim to the test in my first UG class by giving more or less the same final exam I had given a year before for the same course in Waterloo, and the Hong Kong students got nearly exactly the same average grade as their allegedly superior colleagues in Waterloo.

This term, in my first-year course, the students must write several animations, using a very primitive geometric animation language. The point here for the students is not to learn the language, but to slowly get used to thinking algorithmically and to writing small programs. The students seem to have a lot more fun writing these animations than implementing boring data structures as they would normally do in this course (see: http://www.cs.ust.hk/faculty/rudolf/Courses/DS01_f/Animations/index.html). In their animations they show a high degree of creativity, and they obviously spend many hours on sophisticated implementations. In particular, their animations compare very well with those produced by Duke University students (see http://www.cs.duke.edu/courses/spring01/cps049s/students/jawaa.html).

So, Hong Kong students are not really dumb after all (compared to their North American fellows), they are just different. They were educated in a school system

whose highest value is memorizing facts. And they literally work day and night—most student emails I receive are sent after 11 pm.

It is our task at the university to teach students to use their own brains to obtain a solution to a problem, and to emphasize that it is not possible to just memorize everything. This task requires teaching methods different from just lecturing at the whiteboard (or using slides or PowerPoint). To engage the students they must be challenged in the classroom, in the tutorials, and in their homework assignments. When this is accomplished, then teachers (and students) will have more fun than frustration.