

Teaching Statement

Yegor Bryukhov

Goals/Objectives

I believe that different courses set different goals for the instructor who teach them (some goals are always the same – like understanding of the core ideas behind the course). For example, Calculus course, being taught to engineers, first of all have to develop skills of applying certain techniques (differentiation, integration, etc), and we want those skills to retain even as time passes. On the other hand, Algorithms course is not necessarily about memorizing the particular algorithms but more about fundamental concepts and practical skills. It is not necessary for a student to remember those particular algorithms several years later, but they should still remember what is a hash-table, how to estimate complexity of an algorithms, etc.

Means/Methods

As we know, the only way to retain some skill for a long time is by practicing in it for a long enough time. Even if a student understood something very well but did not use it much enough, (s)he will most probably forget it. So the practical exercises have to be chosen according to the long-term objectives of the course at hand.

In my opinion, for computer science courses both pencil-and-paper exercises and programming assignments are equally important. I usually return the incorrect code for students to fix it and resubmit.

I do not believe in group/collaborative projects. They could be very effective but there are too many factors that can easily spoil the whole idea of collaborative learning. At least I do not know how to avoid slipping to the situation with one leading person and all others simply drifting with the current.

I also think that big (long) projects with no intermediate check-points are dangerous. It would be better to split such a projects in a sequence of relatively small subprojects. Then students' performance could be evaluated independently for each subproject and lagging (under performing) students could use reference implementations of already completed steps.

Teaching Experience

I taught a variety of courses from entry level “Introduction to Computer Science and Programming” to intermediate Calculus I and Algorithms; assisted in “Programming Languages Paradigms” and graduate level “Logical Foundations of Artificial Intelligence”.

It might not be a good idea to discuss particular students in the next paragraph

In the “Introduction to CS” course, which was targeted for students majoring in engineering, I had two students majoring one in music and another architecture, they misunderstood the course description and realized their mistake too late for painless withdrawal. In the same class I also had a student with a learning disability. All three of them actively attended my office hours and worked hardy, as a result they successfully passed the final. It was really aspiring to help them with their achievements.

In my work I use a very simple webpage to post all the announcements, syllabus, homework and project assignments, submission requirements and solutions. I also actively communicate with my students via e-mail.

My typical class starts with discussion of the homework and review of recent topics. Then I present the theoretical part of new material, its connection with previous topics. Then follows several kinds of typical problems and how they can be solved. I also present the fragments of code if it is relevant.

Problems

The main problems my students seem to have are the lack of abstract thinking, weak problem solving skills, and reluctance to test and debug the code they submit. Unfortunately, I came across these problems with undergraduate students of all levels.

The inability of an average student to solve non-standard problems is understandable, such a skill has to be specifically trained for years. But I have to admit that I myself lack the experience to cultivate this skill in an audience, which is not willing to be puzzled with non-standard problems.

The reluctance of the students to test and debug the code they submit – is probably the fault of City College CS Department Curriculum at least to some extent. It has no course that would touch the topics of testing, debugging and quality control. It seems, that the department thinks that those are the natural abilities of students that enroll to the program.

Another lesson, I have learned, is that under performing students need a constant feedback on their progress, otherwise they think that a couple of failed tests is just a coincidence, and that they will catch up on the final. But this is rarely a true belief.

Currently, as one of my fellow classmates put it, “every test feels like a personal failure”, as my expectations about results happen (?turn out?) to be higher than the actual performance of my students. I believe that there are two reasons for it - my relative lack of teaching experience, and the fact that I did not have a luxury of teaching the same course more than once.

What also worries me is that some students think that they bought good grades, when they paid the tuition. And that by giving them a bad grade I “violate the contract”. This situation is pitiful, especially in the context of unhealthy litigious mood in the country.

The last sentence should probably be omitted.

Measuring My Effectiveness

Well, I do not have anything to say here really.

Why Teaching?

I don't really have anything to say here.