Re-engineering the value proposition for class attendance in the digital age

Alva L. Couch
Tufts University

Session: Tools, techniques, and best practices of engineering education for the digital generation

Introduction

Class attendance is down. When asked why they do not attend class, students point out that the textbook has all that is needed, and the web is full of helpful material, including the course lecture notes, materials for similar courses, and even video lectures recorded by other instructors! The traditional class lecture format repeats what is accessible by many other means, and students often report on course evaluations that their real learning occurs in lab exercises or when doing homework. Why should students come to class?

We study a very simple way to improve the value proposition for attending class in an age of ubiquitous information availability on the internet. Rather than being a source of information, a classroom is conceptualized instead as a *social nexus in which valuable social interactions occur* between professor and students, as well as student-to-student. We reinforce the social value of the classroom experience via small changes that are practical to implement in existing courses, improve social immediacy between instructor and students, reinforce course outcomes and objectives, save instructor time, and generally improve the perceived value of attending class.

The value of social interaction

There is a wealth of strong evidence in the education literature that social interaction and discussion improves learning. Teaching works best when it is considered a social activity rather than simply imparting knowledge (1). Community building is a key part of this socialization of teaching (2). Social interaction better prepares students for their eventual roles in the world, and teachers should rethink their roles, becoming coaches rather than imparters of knowledge (3). Even reading the textbook is more productive in groups of two students (4). Recently, there has been much work on fostering group-based learning on the internet, for example, in (5). Working in groups is not just a proven mechanism for aiding learning, but also, a key job skill, as well as an objective of ABET in accrediting Engineering programs (6).

Related work

It is well established that discussion and contextualization improve student learning and performance. So- called "community learning" (7) is a matter of creating within the classroom a community with shared values and purpose, a technique that – while it is less prevalent in engineering – is common in liberal arts teaching.

What we do to foster social interaction is very simple. It is much less structured than what has become known as "Collaborative Learning" (8) (9); group sessions are not strictly monitored; in-class interaction is observed, but there is no attempt to enforce equal participation by all members. At the same time, the classroom activity is more structured than Feldman et al's proposed active participation exercises (10), which last no more than 3 minutes at a time and do not result in a deliverable. Our exercises are thus somewhere between "Collaborative Learning" and unstructured active participation exercises in complexity and implementation; they require a tangible product in the form of a completed worksheet, and thus give a sample of how students are doing, but are not used to assess student mastery of the material like, e.g., an exam or regular homework assignment.

A similar strategy to ours is Dimeral's "workbook strategy" (11), in which textbook and lecture are augmented with a "workbook" that is brought to every class and is a source of exercises. Our strategy is much less structured and – because of the immediate feedback – can be tailored more to immediate student needs than the workbook; each exercise can be crafted based upon the results from the last one.

How it works

The class workflow is relatively simple. Lectures are prepared first, and then an exercise is prepared for each lecture and printed on paper. This paper worksheet is handed out to every student[1] after the first 45 minutes to one hour of class, 15-30 minutes from the end of class. Students are broken up into 4-5 person groups according to where they are sitting in the class. We collect one sheet of paper results per group at the end of class, grade them, and scan graded papers into Portable Document Format (PDF) and return it via the class website. We return examinations the same way. We use Adobe Acrobat® to embed detailed comments into the scanned files after scanning, when appropriate. Scanning and posting graded assignments to the web is relatively unskilled labor that can be delegated to a paid student.

Designing exercises

In prior work, classroom activities have proven effective in reinforcing learning in many contexts. However, for the most part, classroom activities have been similar to homework, suitable for doing alone, and have been used to assess proficiency. In considering the classroom experience as fostering a social nexus, our classroom exercises have evolved over several semesters to be very different than homework; they encourage social interaction and reinforcement rather than serving as proof of proficiency. A properly designed classroom exercise can evoke a sense of wonder, excitement, and mystery, and can make a better argument

[1] We initially tried handing out one worksheet per group, but rather quickly discovered that the group could not orient itself to effectively read the worksheet! Thus we always hand out a duplicate sheet to every group member, and then collect only one copy per group, that is intended to represent the group's consensus.

for the value of a subject than listening to a lecture or just knowing how to solve common problems.

The design of our classroom exercises is based upon several unusual and perhaps controversial principles. The role of the exercise is to reinforce concepts and contextualize them in a broader engineering context. The point of an exercise is to give students a **successful experience in applying knowledge,** either alone or with the assistance of the group and the instructor. Thus **it is much more important that students succeed than that they do it by themselves.** The role of the instructor is to foster a spirit of cooperation and help groups to succeed with the exercise. The instructor and teaching assistants are an active part of the exercise; they observe, comment, guide, and intervene when students get onto the wrong track. Another role of the exercise is to reinforce how knowledge fits into the overall picture of engineering. One exercise sheet can contain exercises of varying difficulty, from straightforward answers to research-level questions requiring knowledge from outside the course context.

An example exercise

As an example, consider the following exercise taken from Cloud and Power-Aware Computing in the spring of 2010. The lecture concerned the designs for service switching for a data storage service and the design tradeoffs of each kind. The exercise questions included:

1. List the disadvantages of flow-based switching that motivate a flowless switching model.

2. Draw a time-space diagram of a service that makes two queries into the datastore, and mark the parts that service switching can control.

3. In the (strongly consistent) AppEngine datastore, writes don't block. Draw a time-space diagram of a service that only writes data into the datastore and does not read anything back from the datastore. What parts of this diagram can be controlled via service switching?

4. Is an array of 10 server instances with flowless service switching about "twice as fast" as an array of 5 server instances that host the same services? Why or why not?

5. (Advanced) Suppose that you have a datastore that exhibits *eventual consistency*, combined with a *flowless* service switch. If the *datastore* is not strongly consistent, how can the *application* be written so that subsequent interactions with the datastore (e.g., service requests to fetch the results of a prior transaction) retrieve strongly consistent data?

Students were given 15 minutes to complete this exercise in groups. This format evolved over several classes; this example represents our best known practice, taken from the most recent course.

Assessment and Evaluation

This format has been tried by the author in several courses over a three-year period. During that period the method evolved significantly, and the current report centers on three courses offered after the method stabilized into its current form: Operating Systems (Fall 2009), Software Engineering (Fall 2009), and preliminary results from Cloud Computing (Spring 2010).

Software Engineering (Fall 2009; Tufts Comp180) is a senior capstone course required of all students in the “Bachelor of Science in Computer Science” degree in the school of Engineering. The student population included all seniors in the BSCS, plus several seniors in related programs and two graduate students from Electrical Engineering. In this course the in-class exercises were required, graded like homework, and listed in the syllabus as 10% of each student’s grade. Class participation has always been required in this course, but the instructor was teaching it for the first time in many years, and the exercises represented a change in participation format from previous runs of the course. A total of 14 students were enrolled.

Operating Systems (Fall 2009, Tufts Comp111) is required of Computer Engineering students and elective for Computer Science students. Students enter the course from two different prerequisites: EE14 (Microprocessors) and Comp40 (Architectures). In this course, the in-class exercises were instituted mid-course, after an initial quiz indicated serious comprehension problems for a majority of students. Rather than requiring the exercises, the instructor promised each student one point back on the quiz in question for each classroom exercise in which the student participated. Thus all exercises were awarded participation credit, though correctness was checked and indicated on each sheet. The course is a regular offering of the instructor and has been taught by the instructor a total of three times from the same syllabus. Prior to Fall 2009, it had been taught without classroom exercises. A total of 23 students were enrolled, including 6 from Computer Engineering and 17 from Computer Science.

Cloud Computing (Spring 2010, Tufts Comp150CPA) is an advanced elective in Computer Science for a mix of seniors, Masters’ students, and professional developers. In this course, the in-class exercises were refined even further than those for fall 2009. In-class exercises are required and listed as part of the grade, but are graded mostly on participation with three grades: 10/10 for ideal work, 9/10 for minor issues, and 8/10 if there are major misunderstandings. The average score is between 9 and 10. This course is new and experimental, and not expected to become a regular course offering, but rather, to evolve further into a final form that will become a regular offering. As of this writing, the course is in progress and a total of 17 students are enrolled.

Student comments

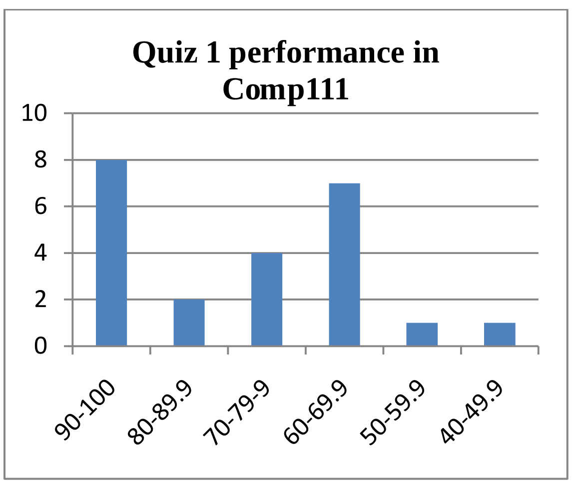
Student comments for Comp111 and Comp180 were collected both midterm via web forms, outside of class, and at the end of the term via paper forms, in class. The unique features of the course were not specifically mentioned in the questions.

Student comments for the courses were unusual in not singling out the classroom exercises as either positive or negative. While a very small number of students wrote that the exercises were useful, there was no overwhelming consensus. In Comp111, which has been taught from the same syllabus by the same instructor three times, there was surprisingly little difference between student attitudes about the course (as reported on the forms) before and after the change.

Separate from written comments, several students commented verbally in person to the instructor about the exercises during office hours. A female student admitted to being initially “nervous” about participating, but also indicated that she got over that after a couple of classes. Surprisingly, one top student admitted to having trouble dealing with the immediacy of the exercises to the lectures, and preferred more time to absorb material before being put on the spot to apply material, while another top student considered the exercises to be too easy.

Grade distributions

One thing that did change substantively in Comp111 was the grade distributions on quizzes before and after the change(Figure 1). The in-class exercises were not planned in advance for Comp111, but were instituted as an emergency measure because of surprisingly low first quiz scores. On the second quiz, of comparable difficulty, students did much better. Though the sample size is too small for any conclusions, this suggests that structured in-class practice for quizzes is more effective at improving quiz scores than giving students a review sheet on which to practice outside of class, with answers.



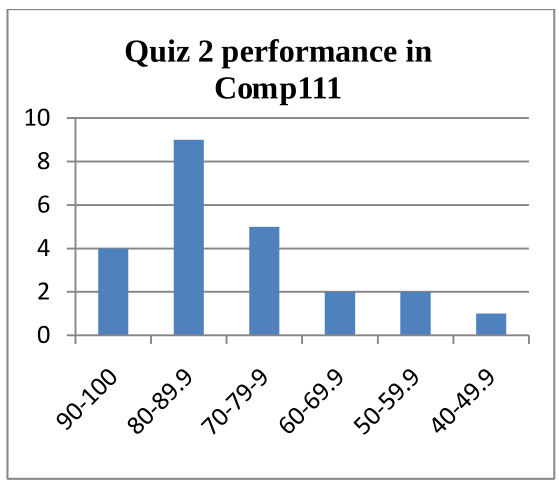


Figure 1: Comparative “before-and-after” quiz performance in Comp111

Class attendance and participation

Operating systems had suffered from a lack of class attendance in previous runs of the course. Out of 22 students in this run of the course, only two had serious attendance problems, as shown in Figure 2. The mass absence on the third-from-last exercise was an excused absence due to an on-campus activity.

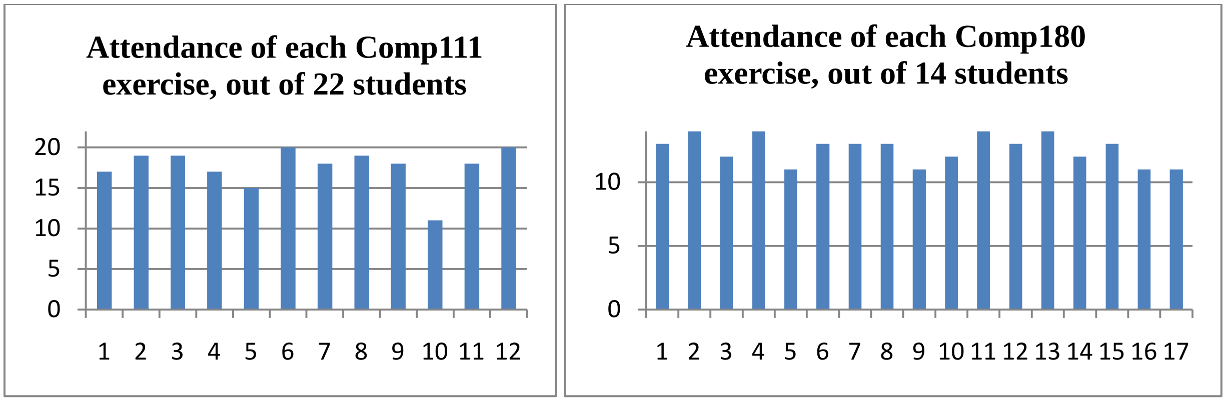


Figure 2: Exercise attendance patterns in Comp111 and Comp180

In Software Engineering, attendance was specifically required in the syllabus, but except for two serious attendance problems in Comp111 (both from students who intended to drop the course), the attendance rolls look quite similar. Attendance patterns did not seem to depend particularly upon whether the exercises were "extra credit" or "required", or whether exercises were graded for correctness or just attendance.

One notable "before and after" effect was that in-class questions increased dramatically after exercises were instituted in Comp111. External visitors observing the class as part of a science education class were astonished at the overall involvement of students in the exercise, as well as almost universal participation in the group exercise (with one exception: a student who planned to drop the class, who also had problems attending class).

Social immediacy

The social immediacy of the class was measured by tracking whether students subscribed to twitter and whether they clicked on twitter URLs in tweets, which were tracked via the tracking built into the URL shortening service bit.ly. Surprisingly, only about ¼ of students actually subscribed to the courses' twitter channels, though the click-through rate indicated that more students (approximately ½) were using the tweets than were subscribed. Most surprising, there was a high click-through rate from unknown users in Europe, indicating that the course materials were being used by people outside the courses. By comparison, another course at the freshman/sophomore level (Comp20: Web Programming) reported almost 100% twitter subscription rates. We attribute this discrepancy to the fact that all of our materials were present on the class website, so that twitter was not strictly necessary, while in the other courses, twitter was the sole source of some kinds of information. Most surprising, **there were no comments – either positive or negative – about the use of twitter in written student comments**. This is some evidence that the majority of students in this class preferred to get their information from the class website, instead of from twitter, **when given the choice**.

Some surprises

Some surprising things about this class format did not show up in student comments, or behavior patterns, but rather, in how the format affected course materials by providing a new kind of direct feedback. The effect was most obvious in Comp111: Operating Systems, which had been taught from the same syllabus twice before by the same person. The practice of creating and assigning a classroom exercise for every lecture had a profound effect upon lecture content, class focus, and student performance data available to the instructor. In comparing archived course notes from before and after the change, lecture notes from before the change were often relatively unfocused and lecture objectives were intangible. Adopting the class format forced me as a lecturer to identify tangible lecture objectives and outcomes (as themes for the ubiquitous exercises) and then tune the lecture notes to emphasize those objectives. As a result of the ongoing and immediate feedback on student performance after every lecture, *course materials and structure dramatically improved in pursuing clear learning objectives and outcomes.* This was in part due to my belief in what is called "teaching as learning" (1); a personal commitment to learn what students need in concert with their learning what I wish to teach them.

Conclusions

This technique surprised us in many ways. A simple technique can bring about big changes. It can reduce instructor workload rather than increase it, while increasing instructor feedback, clarity of course outcomes, student performance, and student excitement. I knew I was doing something "right" when students – several times – did not want the class to end so that they could continue to tackle the advanced problems. It is not as formal as prior techniques, and evidence is that it does not need to be. In a world where both student and professor are always short on time, it seems to make more valuable use of that time. I think that this test of time – more than anything else – will determine the future of the in-person classroom.

Biography

Alva L. Couch is an Associate Professor of Computer Science in the School of Engineering at Tufts University. He has taught at Tufts for over 20 years, and received the 1996 Tufts Leibner Award for teaching and advising, the 2002 Tufts Fischer Award for engineering teaching, and the 2009 Tufts Alumni Service Award for contributions to the Tufts community. As a researcher in the developing discipline of System Administration, he has received two best paper awards in the Large Installation System Administration (LISA) conference(1996 and 2005), one LISA best student paper award (2000), and the 2003 System Administrators' Guild Professional Service Award for contributions to the theory of system administration. He served as LISA program chair in 2002, and currently serves as Secretary of the USENIX Board of Directors and chair of the LISA conference steering committee. He can be reached by email to couch@cs.tufts.edu.

Bibliography

1. *Teaching, as Learning, in Practice.* **Lave, Jean.** 1996, Mind, Culture, and Activity, Vol. 3(3).

2. *Enhancing Learning by Community.* **Tinto, Vincent.** s.l. : NEA Higher Education, 1997, Thought and Action , pp. 53-59.

3. **Denning, Peter.** Educating a New Engineer. *Communications of the ACM.* December 1992, Vol. 35(12), pp. 83-97.

4. *Cooperative Dyads: Impact on Text Learning and Transfer.* **McDonald, Barbara, Larson, Celia and Danserau, Donald.** 1985, Contemporary Educational Psychology, Vol. 10, pp. 369-377.

5. *A Flexible Framework for Online Collaborative Learning.* **Redmond, Petria, Lock and Jennifer.** 2006, Internet and Higher Education, Vol. 9, pp. 267-276.

6. *Designing and Teaching Courses to Satisfy the ABET Engineering Criteria.* **Felder, Richard and Brent, Rebecca.** 2003, Journal of Engineering Education, Vol. 92(1), pp. 7-25.

7. *Building Classroom Community through Group Assignments, Role Plays, and Modeling.* **Smith, Zerec K.** Boston, MA : s.n., 1998. 1998 Annual Meeting of the American Political Science Association .

8. **Adams, Dennis and Hamm, Mary.** *Cooperative Learning: Critical Thinking and Collaboration Across the Curriculum.* Springfield, IL : Charles C Thomas, 1990. 0-398005644-7.

9. **Felder, Richard and Brent, Rebecca.** Cooperative Learning. [ed.] P. A. Mabrouk. *Active Learning: Models from the Analytical Sciences.* Washington DC : American Chemical Society, 2007, 4, pp. 34-53.

10. *The Future OF Engineering Education II: Teaching Methods that Work.* **Felder, Richard, et al.** 2000, Chemical Engineering Education, Vol. 34(1), pp. 26-39.

11. *Effective Teaching and Active Learning of Engineering Courses with Workbook Strategy.* **Dimeral, Yasir.** Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition.

12. *Learning and Teaching Styles in Engineering Education.* **Felder, Richard and Silverman, Linda.** 1988, Engineering Education, Vol. 78(7), pp. 674-681.