Abstract – We describe an approach to teaching mid-sized Computer Science classes in a highly personalized manner. We began with an enumeration of fundamentals that reflect commonly recognized benefits of tutoring: student accountability, customized instruction, clear expectations, emphasis on discussion, and regular evaluation and feedback. For classes of 75 or more some of these appeared unattainable without an army of teaching assistants. But we succeeded, experimenting first in a senior level class of 23 with a design that has remained surprisingly stable as we have transitioned it into discrete math classes of 75 to 80. Our approach includes a novel mix of old and new teaching tactics and techniques combined to create a discussion oriented, high-feedback, personalized, group learning environment. We evaluate the costs and benefits associated with our approach, including the surprising result that individual, video-taped exit interviews add little to the instructor's and TA's burden, when all costs and benefits are evaluated.

Index Terms – Tutoring, personalized instruction, scalable instruction methods, oral exams, discussion.

INTRODUCTION

An abundance of literature attests to the effectiveness of tutoring, particularly as contrasted with the lesser effectiveness of lecturing [2], [5]. Aware of this contrast in effectiveness, the authors embarked on a teaching experiment, admittedly somewhat naively with respect to the existing body of pedagogy literature. What the authors did bring to the experiment, particularly the lead author, is decades of experience with more conventional and less effective teaching methods and outcomes in an assortment of Computer Science classes. This paper describes the teaching experiment and its very encouraging results.

There are three critical properties of tutoring that we believe lead to its effectiveness: 1) customizable learning, 2) regular evaluation and feedback and 3) student accountability. Customizable learning occurs because of the significant role discussion plays in the tutoring process. A tutor who, primarily through discussion, identifies a student’s needs and who can detect when he or she is encountering difficulties, can customize lessons accordingly. Regular evaluation and feedback occur, again, mainly because of the role of discussion in tutoring. In the course of discussing material and evaluating a student’s understanding, a tutor can identify and report on perceived difficulties, providing the student with early feedback. Evaluation and feedback also allow a tutor to communicate expectations. Student accountability occurs as long as the tutor requires the student to demonstrate proficiency. In order to meet expectations, the student must behave in an accountable manner, both during instruction, and during private preparation and review. The potential embarrassment associated with a lack of preparation is motivating.

As a result of experience prior to our experiment we have assumed extreme positions on two pedagogical variables: setting expectations and emphasizing problem solving. Our positions integrate nicely with the incorporation of tutoring properties into medium to large classes, as we establish later. With respect to setting expectations, we advocate posting a database of questions and related material, along with relatively correct timetables for discussion, in their entirety before the semester begins. Further, we advocate that the database of questions define the only material that will appear on exams. Problem solving invites student involvement and provides an opportunity to measure their understanding. By posting a database of questions of a problem-solving nature, one can focus the semester’s effort on problem solving and discussion. By drawing exam questions only from the database, one can take a significant step towards setting expectations early and completely.

In the design of our teaching approach, we identified eight key decisions, including those on expectations and use of problem solving, that collectively created an environment in which the properties of tutoring could be realized in a conventional classroom setting. For each of the decisions we established how it related to tutoring properties. The design represented by these decisions is scalable; we have confirmed that it works in classes up to size 80. We argue the design can scale to larger class sizes with little loss in effectiveness.

Our approach has been applied seven times in two classes, one a senior-level course on programming languages (twice) and the other a beginning course on discrete math (five times). In all seven instances student evaluations have been higher than evaluations in those courses when our approach was not used. We do not have a reliable measure of the effect of our approach on learning quality or retention other than anecdotal evidence. We plan to measure them in future courses.

1 Paul F. Reynolds, Jr., Professor, Dept of Computer Science, University of Virginia, reynolds@virginia.edu
2 Chris Milner, Assistant Professor, Dept of Computer Science, University of Virginia, cmilner@virginia.edu
3 Timothy Highley, Dept of Computer Science, University of Virginia, tihighley@virginia.edu
**Design**

Cognizant of critical properties of tutoring (customizable learning, regular evaluation and feedback and student accountability), and with a strong bias in favor of setting expectations and focusing on problem solving, we identified eight important course design decisions: 1) use of a database of questions as the primary source material for classroom and exam use, 2) intentional focus on course material and away from the instructor, 3) use of a webpage for broadcasting information between class sessions, 4) under 12 hour response time to student initiated inquiries, 5) emphasis on problem solving, 6) frequent use of discussion groups during class, 7) group-oriented, unpledged, homework assignments, and 8) a video-taped 20 minute oral exit interview for each student.

We explore the relationship between the decisions and tutoring properties after discussing the decisions themselves.

*Database of Questions*

Our goal was to construct a database of questions 1) comprehensive in coverage, 2) sufficiently varied in degree of difficulty to permit students to gauge their level of understanding of course material, 3) representative of the weight given to course topics and 4) sufficiently large to discourage memorization. This database should be completed and posted before the semester begins so that an instructor can identify it in an opening presentation that includes discussion of expectations: “If you can master the questions in this database you will possess all of the knowledge required of a Discrete Math I student.”

Two questions that often arise are: “Aren’t you teaching to the questions?” and “Can’t the students just memorize answers?” Without forethought in course design, the answer to either question could be affirmative but with some care never should be. The questions should be used to set expectations and provide a source of small tests that can be employed to evaluate preparation for addressing more interesting problems. For example, students likely to take a discrete math course are generally aware of the importance of encryption and are interested in its inner workings. So knowledge about encryption can be the motivator and questions that cover material on functions and prime numbers can be cited as representing the skills required to better understand encryption. The questions themselves become secondary.

Regarding memorization, one can employ both prevention and redirected motivation. A database with hundreds of questions, each of which may be presented (e.g. on a exam) with small changes in constants, variables or problem size, reduces the advantages of memorization to negligible levels. The lead author once received this comment on a student evaluation: “300 questions is too many to memorize!” which elicits two responses: 1) the student missed the point and 2) the argument here is affirmed. Redirected motivation relates to associating subsets of the database of problems with interesting applications, such as encryption. Students learn they must understand what the problems are teaching to grasp the essentials of encryption. Memorization won’t do.

Finally, the database must not be larger than necessary. More questions is not necessarily better! The instructor should take care to not overwhelm students with extraneous material. The questions should capture the material the students need to master and no more.

*Focus on Material*

A recent study out of the University of Texas shows that performance on student evaluations is strongly related to instructor attractiveness [3]. Clearly it’s easy for a course to become too much about the instructor and not enough about learning. A course designed around clearly established expectations, including the questions database we advocate, allows an instructor to become a collaborator, and less likely the center of attention. From a student’s perspective motivation lies in applications (understanding encryption, writing correct programs, winning at poker) and the challenge becomes mastering essential material. For the instructor who understands the students’ motivations, the material becomes the challenge that all participants, including the instructor, can address together. With the addition of group homework assignments, extensive amounts of classroom discussion, and quick responses to student questions through electronic communications, the collaboration can work.

*News Broadcast*

Interesting events occur between class meetings. A course built around discussion requires a means for communicating in the interstices. Before the era of Wikis [1] (interactive webpages) a webpage maintained by the instructor or a TA was the best alternative. Material for the page can range from pedestrian, e.g. corrections to assignments, and clarifications for classroom discussions, to novel, e.g. pointer to a newly created website describing a just discovered proof for Fermat’s last theorem.

*Timely Response to Individuals*

Communication by email has entered the core of many university-level courses. As with a broadcast method of communication, a class oriented around discussion requires individual support that is available between class meetings. Email and instant messaging enable the required types of communication. Monitoring communication channels and ensuring timely response are necessary for providing feedback that encourages self-initiated problem solving.

With the advent of Wikis, both broadcast and more customized response can be supported. However, for the student seeking the comfort of more private communications, email and instant messaging remain a preferred communication option.
Problem Solving

A Computer Science undergraduate education focuses largely on problem solving skills. Issues related to correctness of solutions and quality of solutions are separable. Testing for correctness can often be delegated to individuals or peer groups and be carried out outside of the classroom. Analysis of solution quality is often best left for in class discussion. In both cases the role of discussion is primary, and it invites peer involvement. Because of the importance of discussion to the success of tutoring we chose to emphasize it in our approach as well by exploiting the high degree of problem solving (and resulting opportunities for discussion) in Computer Science.

In addition, problem solving offers an opportunity for students to gauge their own progress. If the course expectation is that by a particular time in the semester the student is supposed to be able to solve an identified subset of the database of problems, a student, or better a small group of students, can conduct fairly reliable analyses of their ability to do so. For those problems they cannot solve, peer discussion, email requests for help and in class discussion all offer opportunities to remedy perceived deficiencies.

Discussion Groups

Discussion creates opportunities to customize learning and offer feedback. Peer pressure and the opportunity to perform well in class (and avoid potential embarrassment) address student accountability. To keep students involved, discussions should take place in groups of varying sizes ranging from two individuals to the entire class. Students must provide a significant portion of feedback, particularly in smaller groups. Discussion must lead to understanding, and not just an answer to a specific problem.

Students must overcome fears associated with discussion. Working in small groups early on, observing other students experience varying levels of success in classroom performance and believing the classroom is a non-threatening environment are all necessary to the success of a discussion-oriented course. To encourage discussion and overall participation we determined that the instructor must learn student names, the earlier the better. A smile and use of the student’s name often help a student overcome fears that arise in classes where they might otherwise feel anonymous and detached.

Group Homework

A teaching approach that emphasizes problem solving, discussion and frequent feedback, benefits from unrestricted assignments. Students should be allowed to request help from others, just as they are encouraged to approach a TA or the instructor. Discussion, particularly in the classroom, fosters the formation of groups who work well together. These groups should be allowed to operate unconstrained in seeking solutions to problems they face in common outside of the classroom.

Concerns that often arise are “Won’t weaker students lean too heavily on more capable students?” and “Won’t students cheat?” These questions are essentially the same when collaboration is permitted. Yes, weaker students (e.g. with respect to course material or time management skills) may very well derive unfair advantage from association with smarter or more organized students. However, they should know from the first day of class onward that this sort of benefit is double-edged. If, in the aggregate, exams count for a considerably higher portion of a final grade than homework, then copying homework assignments has few benefits and considerable cost. Students who do not appreciate this initially generally learn it through classroom discussion or during an exam.

Oral Exit Interview

The thought of soloing for 20 minutes at a whiteboard, in front of an examiner with a video camera, is sobering. Students should know from the beginning of the course that they will be required to participate in an exit interview. They should be encouraged to integrate it into their entire approach to the course (See “Won’t students cheat on homework?” above). They should be encouraged not to fear it as long as they have met expectations during the semester. The course should make the interview nearly effortless.

By design the exit interview provides focus: “Student, you will need to complete this demanding task. I am providing you with 1) well defined expectations, as captured in a database of questions, 2) extensive opportunities, and strong encouragement and incentives to discuss course material, and 3) a hierarchy of individuals (peers, TA, instructor) available with relatively short response times to answer your questions. Use the opportunities provided to you and you should do well in the oral exit interview.”

A course with an oral exit interview becomes a metaphor for professional life. Oral interviews are the norm once a student leaves an undergraduate environment, whether their next station is in a graduate program or fulltime employment. Many students recognize this and are grateful that the course helps them to prepare for more than simply learning the course’s intended materials. An additional benefit that was initially underappreciated by the authors is a precipitous drop in matters involving dishonorable behavior. A student who knows s/he and all peers will be evaluated in a manner which essentially precludes all cheating has less incentive to cheat.

Design Decisions and Tutoring

Table I displays the relationship between each of the design decisions and the critical properties of tutoring. An X appears wherever the corresponding decision is believed to support the corresponding tutoring property. Thus news broadcast supports regular evaluation and feedback (only). While none of the tutoring properties is supported by all of the decisions, each of the properties is supported by a majority of the decisions.
Does our design add up to the equivalent of tutoring each student? No, and we don’t claim it. However, we do ensure that discussion plays a significant role, students have a clear idea of expectations and where they stand with respect to them, and they have numerous opportunities to customize their instruction in accordance with their needs and expectations. Students are held accountable because they have constant access to what they should know (database) and frequent discussion, homework and the exit interview all test for compliance.

Courses built using our design decisions can be taught with the usual resources (for us, one instructor and one TA). Can the benefits of tutoring be realized with fewer resources? Are there alternative decisions that would lead to lower resource usage and/or improved delivery of tutoring’s benefits? We leave these questions for future exploration.

**IMPLEMENTATION**

We have implemented our design in seven instances of two courses. The first, a senior level programming languages course, was where we carried out most of our prototyping. In both instances of the course there were under 25 students. The subject material was a mix of programming language design, and implementation. Both classes were taught by the same instructor (Reynolds) with one TA (Highley). The second, Discrete Math I, has been taught five times (four by Reynolds and once by Milner), with Highley as a senior TA twice. Because our approach to all seven classes has remained relatively constant, we describe implementation with respect to a semester timeline.

**Before a Course Begins**

The database of questions must be constructed in accordance with the design requirements listed for it in the Design section. A database of 300 to 400 questions works well. We have found course texts excellent sources of questions.

The lead author constructed the programming languages database from questions gathered over years of teaching the course. In the second instance of the course, author Highley was supported by a University of Virginia (UVa) grant to build an improved database. His effort spanned a summer and produced a much improved version. For Discrete Math the lead author constructed a database of questions using the course text as the primary source of material. During the five times this course has been taught the database has been refined at a rate of about 10 to 15% per course instance, by choice. The Discrete Math database currently contains about 400 questions and has been pared on more than one occasion to ensure the material does not overwhelm students.

The questions need to be separated into three groups: those that are assigned for homework, those that are the topic of in class discussion, and those that are added to round out the list of potential questions for exams. These groups should be clearly marked before the semester begins.

**First Day of Class**

On the first day, students are introduced to the unusual characteristics of the course: 1) all questions for homework, discussion and exams are posted on the web and will be the only source for exam questions, 2) unpledged (groups allowed) homework assignments, 3) strong discussion and problem solving orientation and 4) an oral exit interview is required of all students at the end of the semester.

Students are told they will be individually videotaped in the next class, so that the instructor can learn their names and something about them.

The TA is introduced and is encouraged to mingle as the instructor continues an overview of the class.
A homework assignment is presented in a form that is used repeatedly throughout the semester: 1) reading, 2) in class problems to prepare to discuss, and 3) written homework problems, with due date.

Second Day of Class

The second day of class is led by the TA. The instructor quietly circulates through the room with a video camera, stopping in front of each student. Students are asked to speak their names and anything they wish to tell about themselves. Identification of nicknames is encouraged. The video, along with other information available through the registration process, forms a class portfolio for the instructor and TA. Printed thumbnails of students, along with names, can be studied and memorized by an instructor at low productivity times outside of class. In one instance of the course, the instructor requested students’ permission to post pictures and nicknames to the class website. Students can then refer to this website to learn each others’ names.

Typical Class

At the beginning of every class, expectations for the next class are announced. This includes reading assignments, class problem assignments and written homework assignments. This information is also posted to the class webpage (since the beginning of the semester).

The thread of the day’s discussion can be captured on electronic slides. However, in both Programming Languages and Discrete Math I we intentionally left almost all problem solutions for presentation on a whiteboard. This fosters discussion and creates opportunities for students to participate in solving problems.

Not all discussion should be led by the instructor. On a regular basis students should be encouraged to break into groups and work through one or a small set of problems. While they are doing this the instructor, sometimes assisted by the TA, can roam the classroom dipping into and out of discussions as needed or desired. Students are quite capable of helping each other and are strongly encouraged to do so. On many occasions we instruct students to turn to their neighbor and discuss a question for the next minute or two.

Groups can be managed formally or informally. We have tried both, with a trend towards informal management. The first three times our approach was applied (twice in PL, once in Discrete Math) students were instructed to break into groups of no more than an instructor chosen size, typically six to eight. They were to designate a leader within their group for the remainder of the day. They were to discuss assigned questions. Discussion groups always met during the first 20% of the class period. The purpose of a group was to ensure its leader understood the problems of the day. When the groups broke, the instructor collected the names of the day’s group leaders, who then were woven into the problem discussions for the remainder of the hour. All students had to be a group leader at least twice during the semester.

In more recent classes we have employed ad hoc calls to break into groups, with the requirement that all students become prepared for subsequent class discussion. After groups finished, all students were candidates for being woven into the problem discussions for the remainder of the hour. Participation can vary from answering a question when called upon to working problems at the board.

Homework

Homework is composed of problem sets drawn from the database. All homework is unpledged and collaboration is strongly encouraged. Homework is graded (TA, and undergraduate graders in larger classes). Often we grade subsets of the workset, but do not reveal which problems will be graded until after homework is submitted. We strive to return homework in no more than two class sessions.

Exams

We have tended towards two written exams in two day per week classes and three in three day per week classes. The last exam occurs on the last day of class and the other(s) are spaced proportionally during the semester. As advertised and promised, all exams are derived solely from the omnipresent database of questions. Exams are closed books and notes. Written exams are always returned the next class session, to foster discussion. An answer key is posted on the class webpage the night of the exam.

Outside of Classroom Communication

We use webpages and email extensively to communicate announcements, clarifications, updates and interesting tidbits. Students are encouraged to post questions. Part of the TA’s assignment is to monitor and respond to electronic message traffic. The instructor participates actively as well. A primary goal behind our electronic communication is to maintain threads of discussion begun in the classroom.

The Oral Exit Interview

Oral exit interviews are video-taped. They must be, to protect both the student and the examiner in potential disputes. We intend to transition to webcams and digital recording in future sessions.

Students sign up for their interviews beginning about the third to last week of the semester. Usually we allocate blocks of time ranging over a period of a week, occurring during final exam week. We use a webpage to manage signups.

Candidate questions for an interview are written individually on index cards. We select about 100 of the questions in our database for use in interviews. Each student is asked questions about their last day written exam, and six questions randomly selected from among the 100. A spreadsheet can be used to randomly select questions and support recording of performance.

When a student enters the room, they are reminded of the presence of the camera and then encouraged to ignore it.
Tapes are never intended to be viewed by non-teaching personnel (and never have been). A whiteboard is available. The student is encouraged to relax, and the interview begins. All questions are presented “on-line” meaning once the student declares s/he is finished (or the examiner does when too much time has passed) the student will not be presented the question again. A question may be addressed later if the student remembers it.

All grades are recorded as the exam progresses. When the exam is over, the student and examiner can discuss performance, time permitting.

Oral exit interviews can be conducted by the class TA, and other qualified graduate students.

**Evaluation of Student Performance**

A typical course is graded as follows: homework 20%, written exams 50%, participation 15% and exit interview 15%. Participation is graded based on instructor evaluation of student performance in class activities (discussion based) and student evaluation of student performance (lower weight is given to this component).

**Evaluation and Discussion**

In all seven instances of the two courses taught, student evaluations have been higher, to significantly higher (greater than one standard deviation), than the average for the same course taught at other times. In all cases evaluations have been higher to significantly higher than evaluations for comparable courses in the UVa Engineering School. Discrete math courses are generally unpopular. We regard the evaluation results as significant. A recent chairman’s evaluation for an instance of Discrete Math I: “Very good results in a traditionally tough course to get good scores. Excellent...” We have anecdotal evidence of the form: “My roommate told me to take this course because you can learn so much in it.” – this from students in a different school who are not required to take the course. We lack better evaluations of the course, primarily because resources have not been available to conduct them. The method is successful if it leaves students better prepared in the subject material and its use than they would otherwise be.

Observations of interest include:

- **What about fraternity files for the questions?** Great! We leave the database posted year-round and students are invited to access it, work through the problems and share the results with their friends. It remains too much material to memorize, and enhanced levels of understanding are welcome.

- **With all of the classroom discussion, does chaos ensue?** No. The instructor has to watch for when most groups have reached the end of useful discussion and reacquire everyone’s attention. An announcement such as “Let’s see what we have now.” almost never fails to refocus attention. Students want to see how well they’ve done.

- **Do students object to randomly selected questions in oral exit interviews?** No, they don’t. Through seven classes and half a thousand oral exit interviews there has not been a single complaint. Of course, the advantage of randomly selected questions is students can discuss their interview with others who have not taken it with no impact on exam quality or reliability.

**Do exit interviews get challenged?** Not a one. We’re surprised. Tapes go unviewed and unused until they have to be rewound for the next round of exams.

**Do interviews require a large amount of time to administer?** Yes, but graduate students seem quite willing to help administer them, so instructor time in interviews is often 20%-30% of what it might be. With a small amount of preparation, graduate students can perform as well as the instructor in administering exit interviews. Also consider: when a student leaves their interview all grading for them is complete. There are no long, paper exam, grading sessions. Because the potential for cheating is reduced to near zero, time spent in honor-related proceedings is avoided. The drop-off in honor-related issues has been profound! The lead author has not initiated an honor proceeding since employing the teaching approach discussed here. See [4] for a discussion of what students require to stop cheating.

**Summary**

Tutoring has significant potential, as established in the literature. One can’t tutor traditional university classes. We have presented an approach we believe approaches the benefits of tutoring in a scalable manner.

**Acknowledgment**

We acknowledge the UVa Provost’s Office for database development support and the purchase of a video camera.

**References**


