Turning Exams Into A Learning Experience

Benjamin Yu, George Tsiknis, Meghan Allen
University of British Columbia
Department of Computer Science
2366 Main Mall
Vancouver, BC
Canada V6T 1Z4
benyu, tsiknis, meghana @cs.ubc.ca

ABSTRACT

Most students see exams as a stressful experience, having crammed as much as they possibly can and hoping that the material they have studied would indeed be on the exams, and the material that they did not have the time to study for, nor understand, would be overlooked by their professors. At the same time, instructors see exams as formal assessments of student learning. Exams are seldom thought of as a learning experience. In this paper, we report our experiences in the use of two-stage exams as a learning experience for the students in two different courses. In a two-stage exam, students write an exam individually, then they rewrite the same exam in collaboration with three or four other students. One of the main objectives is to encourage peer instruction and promote learning even during the exams. Students obtain immediate feedback during the exam, and initial results have shown that this type of exam format produces some positive effects in later assessment.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *Computer Science Education, Self-Assessment.*

General Terms

Design, Experimentation, Measurement, Human Factors.

Keywords

Two-stage exams, learning, peer-instruction, assessment, midterms.

1. INTRODUCTION

Traditionally, learning is thought of taking place only in the classroom, labs, or when students are engaged in their homework assignments, or readings, whereas midterms and final exams are thought of only as student assessments so students can demonstrate what they have learned. Learning is not emphasized during exams, but perhaps takes place as a by-product after the exams. However, learning can take place during exams if the exams are conducted in such a way to foster learning rather than solely focusing on recall of material or concepts learned in preparation for the exams. A shift of focus on learning rather than on testing can also reduce the anxiety most students associate with exams. This paper discusses the use of two-stage exams, which turn the exams into a learning experience, in the midterms of two different computer science courses. We will report the students' comments on the midterms and how their performances on the midterms correlate with the final exam that consists of isomorphic questions as those in the midterms. We will also discuss how the administration of two-stage exams can be improved.

This paper is organized into the following sections. Section 2 provides the background information of the two-stage exam. Section 3 describes the courses which we administered the two-stage exam. Section 4 provides the details of what transpired during the exams. Section 5 includes both anecdotal comments from the students on the two-stage exams as well as analysis of their performance in the two-stage exam and the correlation with isomorphic questions on the final exam. Finally, section 6 is a discussion of the experiments, with concluding comments and future work in section 7.

2. TWO-STAGE EXAM

The idea of a two-stage cooperative exam is that students take the same exam twice during an extended period of time but in different settings. These settings can be individual in the beginning, followed by working in pairs, or collaboratively in a larger group. The goal is to turn these testing sessions into a learning experience.

An example of how this is implemented in a large class for midterm or final exams as reported in [5] is as follows: during the first 30 minutes of the class period, the students take a multiple-choice exam with about 20 - 25 questions in it individually. They hand in the answer sheets at the end of the exam. Then right away, they are given the same multiple-choice exam but with added questions in it, and are asked to work on it collaboratively with someone close by for 45 minutes. They can use books, notes, and other resources. The grade of the exam is calculated based on a weighted average of the first submission (75%) and second submission (25%) of the exam. However, if this grade is less than the grade in the first submission (i.e. from the solo effort alone), then the final score of this exam is based solely on the first submission.

With this simple change in exam format throughout the term, it was reported that there was a large improvement in the final exam scores from a mean of 74% to 80%, based only on the solo part of

SIGCSE Conference 2010, March 10–13, 2010, Milwaukee, WI, USA.

the exam. Although it seems that the collaborative component of the exam may have boosted the final score, a statistical comparison with grades from previous years with no collaborative component in the exams shows that there is no dramatic change in grade distribution. The number of students at the bottom rungs of the ladder is fewer with the two-stage cooperative exam strategy, but there is no increase in the upper rungs.

There is another significant benefit of the two-stage exam, which is the immediate feedback the students receive during the group discussions. It has been reported that the single most powerful influence for student achievement is feedback [1, 4]. But it is not just any feedback. Informative, timely, concise feedback, which the students can actually follow up on, is what really counts. In particular, timely feedback received by students while it still matters to them and in time for them to pay attention to further learning or receive further assistance is especially important for student growth and learning [3]. While the group discussions do not guarantee that the students will come up with the correct answers to all the questions in the exam, the group discussions provide ample opportunities for the students to collaborate in solving the problems from different angles and perspectives, thus fostering an enriched environment for learning.

Given the encouraging results from the research literature and similar experiences of two-stage exams in other courses from other disciplines at our university, we decided to try this exam format in the midterms of two computer science courses offered during the summer sessions of 2009 at the University of British Columbia.

3. COURSES / STUDENT BACKGROUND

This section provides the background information of the two courses where two-stage exams were administered.

3.1 Course 1: Models of Computation

The first course is one of our first year courses on formal models of computation. It covers similar topics as the Discrete Structures knowledge area of the ACM Computer Science Curriculum [2]. The course explores formal modeling systems to understand and explore the capabilities of computers, and, more generally, of any process for solving a problem. Topics include reasoning about formal systems and proving relevant properties, propositional and predicate logic, creating regular expressions, creating and using Deterministic Finite Automatons, justifying behaviors and correctness of some algorithms, etc.

A total of 37 students registered in this course with 11 females and 26 males. The course included three 2.5 hour lectures, four hours of lab, and one hour of tutorial each week for 6 weeks.

3.2 Course 2: Introduction to Software Development

The second course is one of the beginning programming courses for Computer Science majors at the University of British Columbia. It is usually taken after the introductory programming course in the first year. Because of the applied nature of the course material most of its learning goals are at the *application* level of the *cognitive domain* aspect of the learning taxonomy. The course introduces students to the software development process. It covers similar topics as the introductory part of the Software Engineering knowledge area of the ACM Computer Science Curriculum [2]. Topics include design practices, data abstractions, inheritance, testing, concurrency, and distributed computing.

A total of 59 students registered in this course with 24 females and 25 males. The course included three 2.5 hour lectures and two two-hour labs per week for 6 weeks.

4. MIDTERM EXAM

Both midterm exams took place in the lecture theatre. In Models of Computation the students were given 80 minutes to do the test individually, and in Introduction to Software Development the students were given 50 minutes to write the test individually. In both courses after the individual portion of the exam, the students then rewrote the same test in groups of three or four after they handed in their individual work. In both cases, there were significant discussions among the students during the collaborative part of the exam. The following sections provide additional details of what took place in each of the courses.

Models of Computation

The students were given approximately 50 minutes to complete the exam in groups of three or four. The students had been working on problems in small groups during lecture all semester and were therefore already comfortable working with their classmates. They quickly organized themselves into 11 groups. We observed that most of the groups worked through each problem together and discussed the best approach to take for each particular problem. In most groups the students were having interesting discussions and did seem to be learning from one another.

Three of the groups stood out as problematic. The first group was entirely comprised of extremely high-achieving students. Instead of discussing each question, this group split the exam up into four sections and each student completed one section of the exam. They then reviewed each question and handed in their exam early. This was a reasonable strategy for this particular group since each student was strong and able to do quite well without input from the other group members. However, in this particular instance the two-stage exam was not a learning experience for the students, instead it was just a simple re-write of questions they had previously answered.

The second group was comprised of three low-achieving students and one high-achieving student. None of the three low-achieving students were able to provide much useful input so the high-achieving student essentially wrote the entire group exam. This was unfair for the high-achieving student because he greatly assisted the other members of his group and did not receive any reciprocal assistance in understanding the material on the exam.

The third group was comprised of one extremely high-achieving student and three average-achieving students. This group had good debates about the questions. However, the extremely high-achieving student was unable to convince the other three group members that her answers were correct. This was frustrating for her, and demonstrated that some students would not learn as much as they could from their group members.

Introduction to Software Development

To increase participation and active learning opportunities, a number of group activities are normally included in this course. In each lecture, true-false or multiple choice questions were used during the presentation-discussion of a concept and at least one long group exercise was included at the end of the presentation. During these group exercises, students were asked to work in groups of two or three and use the ideas and techniques they have learned to solve a related problem. After the students have submitted their work, the instructor would ask for input from the groups and present the solution on the board. Typical exercises take up 10 to 15 minutes of the class time.

Since the students already had experience with group work, it was anticipated that they would be pleased with the notion of the two-stage midterm exam when it was announced. However, nearly one third of the students had some concerns with the group exam portion when they were told that their exam mark would be a weighted average of the individual and group exam. A number of good students were afraid that the group work would not provide any significant help for them and it might even slow them down. Some of the low-achieving students felt that they may be embarrassed if they could not contribute enough at the group exam. The concerns diminished considerably when the students were informed that the score of the group exam would not be taken into account if it was lower than the score of the individual exam, and the students could choose their group partners.

As stated before, after the students took the individual exam, they formed groups of three or four and were given a new exam paper with the same exam questions which they had to complete in 30 minutes. It is important to note that none of the questions were multiple choice questions. All the questions required a considerable amount of work to derive the answers which were of the form of short (10 to 15 lines) Java programs, short answers, or mathematical computations.

The first significant notable difference between the group exam and the regular group exercise is that every student actively participated in their group. During the group exercises in the lectures before the midterm exam, five of the students in the class never joined any group; they always worked on their own. Even after a number of attempts to convince them of the benefits of group work, they continued to work on their own. However, during the group exam, it was a pleasant surprise to see these five students not only participated in their groups, but also being very active in the group discussions. A couple of them even took the lead to write out the group's final answers on the exam paper. When they were asked how they liked the new experience, four out of the five replied that they enjoyed it very much. Moreover, these four students started participating in groups during the lecture group exercises after the midterm exam.

The rest of the class was similarly involved in active discussions during the group exam. Because the students had formed the groups by themselves, each student was familiar with their team members, and they were comfortable to openly discuss the exam questions. However, we did notice that there were a few students who took some time to find a group for various reasons. Even in those cases, there were significant discussions among the students during the group component of the exam.

There is another notable difference between the group exams and the individual exams in that the amount of working scribbles is more on the group exams than the individual exams, and occasionally, multiple attempts of the solution of the same question are seen on the group exam papers. This is an indication that even though the time for the group test was less than the time for the individual test, there were significant collaboration and effort from the group to solve the questions.

An analysis of the individual and group responses to each exam question reveals some more interesting facts. The test question on which the group performance had the largest average gain was a programming question on how to use certain data structure for a specific problem. It seems that this type of problems benefit greatly from the group collaboration and the fact that an isomorphic question (more details in the next section) in the final exam shows a 50% improvement above the individual score is a strong indicator that the group activity may have helped in the student learning.

There were also two questions in the exam where isomorphic questions have been included in every midterm given for this course. One of them is on the selection of test cases for a piece of software and the other is on class design. These questions are usually open-ended questions as there are many possible correct answers to them. Even though the improvement from the individual to the group score is only 10% for these questions, the improvement on isomorphic questions in the final exam is around 73% and 52% respectively. It seems that these are also the types of question that benefit well from group discussions.

After the midterm exam, the students were asked if they learned anything new from the group exam. More than half of the class indicated that the exam was indeed a learning experience for them. Only three of the 59 students indicated that they were forced to accept an answer without being convinced that it was better than theirs. Finally, all except five students wanted to have a two-stage final exam.

5. DATA ANALYSIS

We have also conducted online attitudinal survey towards the end of the term. Here are some of the results. For the course on Model of Computation, 75.9% of 29 students surveyed responded positively to the midterm, 13.8% responded negatively, and the rest (10.3%) were neutral.

Three isomorphic questions from the midterm were included in the final to see how the students perform on these questions. The final exam was conducted as individual exams and no group discussion was allowed. Surprisingly, there is quite a significant drop in the overall student performance on these questions in the final exam. Overall, there is an average drop of 6% among the three isomorphic questions. One question skews the result quite significantly with a drop of 23%, another question has no change, and the third isomorphic question in the final has a 10% gain.

For the course on Introduction to Software Development, the end of term online attitudinal survey shows that 75.5% of 49 students surveyed responded positively to the midterm, 18.4% responded negatively, and the rest (6.1%) were neutral.

Three isomorphic questions from the midterm exam were also included in the final exam to see how students perform on these questions. Just like the other course, the final exam was conducted as individual exams and no group discussion was allowed. In this case, the result is more encouraging: 72.5% of the students did better in the first question, 51.7% did better in the second question, and 50% did better in the third question.

5.1 Student Comments

As indicated above, most students are positive about the two-stage exams. Here are some of their comments:

"..we got immediate feedback and thus we immediately were able to learn our mistakes. whereas getting back midterm at a later date makes us care less about the midterm material.."

"..When we did the midterm the second time with our partners, they made me realize certain issues i didn't see before. Also, I learned from them certain techniques they use in handling such questions that is very useful to me now. …"

"… I learned alot (sic) from others. I was also able to help others who didn't understand something that I did …"

"…I trhink (sic) that was fun, amazing and very helpful many benefits can come out of that: team work is important, you get to see different point of view and of approaching a problem. It also turns the stress of the exam into something positive. .."

However, there are also negative comments:

"I felt that I couldn`t put 100% of my input because some of my group members were really persistent on their solutions (even though, the answer key indicated, I was closer to the solution than they were). Our score was somewhat similar to my individual grade and so my mark wasn`t boosted as high as some of my group members (because their individual grade was a lot lower) - this is kind of unfair."

"…partnership wastes a lot of time. It helps in boosting the marks, though, which I like. But in terms of learning, less likely to, especially if we have already done it once individually. Better if we have one exam for individual and one or two questions for the partners."

6. DISCUSSIONS

While there are strengths in the two-stage exams, we did encounter challenges in both of our courses.

- Students who are high-achieving don't seem to get much out of the two-stage exam, while low-achieving students do. This is probably due to the imbalance of group members where stronger students have more to contribute than to receive from other members. One possible way to avoid this is to provide other opportunities for students to work with different groups before the exam so they can find a group they are comfortable in working with. Another possibility is to have the instructor assign students of similar GPA to the same group.

- Students who have difficulty in articulating and explaining their answers in the group discussions also do not feel the group exam was beneficial. This may not due to the exam format but more on the opportunities and practice the students need to have in order to communicate effectively their ideas to others.

- Time was wasted during the exam period (in the case of Introduction to Software Development) for some of the students to organize themselves into groups. Students should be given ample notice to plan ahead who they would like to work with before the exam rather than trying to find a group in the last minute. Again, instructor assignment of groups may be a solution.

- One of the student comments relate to the difficulty level of the exam. If the exam is too easy and the questions are too straight forward, it defeats the purpose of the two-stage exam. In order to turn the exam into a learning experience, the questions in the exam should be challenging and the solutions non-trivial.

- In both courses, the same exam was used in the individual section as well as for the group section of the two-stage exam. To foster even more discussions and knowledge transfer, it would be beneficial to add other additional isomorphic question for the group section.

- In both courses, a number of students commented that the time allocated to the group section was insufficient, especially for discussions. As opposed to [5], the exams conducted in the two courses were not just multiple choice questions. Hence, additional time was required to write out the solutions.

7. CONCLUSION AND FUTURE WORK

The initial feedback from the students as well as their performance in both the midterm exams and the final exam suggest that two-stage exam can be a valuable learning experience. As noted in one of the comments from the students, the two-stage exam "turns the stress of the exam into something positive". There are also some indications that the students do better in the isomorphic questions in the final exam given their experience in the midterm exam. Whether this is due to the learning gain from the two-stage exam or not will require further investigation. At the same time, such performance gain is not evident in all the isomorphic questions. We intend to analyze these questions at greater depth, as well as the student solutions from the solo portion of the two-stage exam, the solutions from the group portion of the two-stage exam, and the solutions from the final exam. We are also conducting two-stage exams at higher level courses of our computer science program. If students see exams as part of the learning process, we hope that their retention of the material will also be greater in later terms, and we plan to conduct longitudinal study that will follow up on this study.

8. REFERENCES

[1] Black, P. & William, D. (1998). Assessment and classroom learning, Assessment in Education, vol. 5, no. 1, pp 7-74.

[2] Computer Science Curriculum 2008: An Interim Revision of CS 2001. (December 2008). Association for Computing Machinery. Retrieved on September 7, 2009 from http://www.acm.org//education/curricula/ComputerScience2008.pdf.

[3] Gibbs, G. & Simpson, C. (2005). Conditions Under Which Assessment Supports Students' Learning, Learning and Teaching in Higher Education, Issue 1, pp. 3 - 31.

[4] Hatte, J.A. (1987). Identifying the salient facets of a model of student learning: a synthesis of meta-analyses, International Journal of Educational Research, vol. 11, pp. 187-212.

[5] Yuretich, R., Khan, S., Leckie, R., Clement. (March 2001). Active-Learning Methods to Improve Student Performance and Scientific Interest in a Large Introductory Oceanography Course. Journal of Geoscience Education. 49(2), p 111- 119.