INVESTIGATING STUDENTS' PERCEPTIONS OF FAIRNESS FOR COMPUTER BASED EXAMS

BY

CHINEDU ALEXANDER EMEKA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Adviser:

Professor ChengXiang Zhai
Associate Professor Craig Zilles

ABSTRACT

Computer-based tests have numerous advantages over paper-based ones, including allowing instructors to scale up their classes by removing the bottleneck of manual grading, as well as facilitating data collection and analysis. At the University of Illinois, instructors have adopted and extended computer-based testing to support asynchronous exams. Students may take their exams asynchronously at convenient times, and they are assigned questions randomly from a pool of question variants. Previous work has been done to compare variants and test equality, but little research has been conducted to determine students' perceptions of fairness of these exams. In this work, we investigate students' fairness perceptions for major aspects of the computerized exams. We perform a qualitative study and lay the groundwork for future quantitative analysis.

To my Grandmother Regina Chiegboka and my Aunty Ngozi Nchekwube.

ACKNOWLEDGMENTS

I would like to thank Professor Zilles for his guidance throughout this project. Professor Zilles is an excellent mentor, and I was fortunate to have the opportunity to work with him. I would also like to thank Professor Zhai and Professor Koyejo for their support throughout my graduate studies; and Professor Geoff Challen, for helping me to become a better instructor during my teaching assistantships in CS 125 under him.

Many thanks to Viveka Kudaligama in the Academic Office for advice given during challenging times in the program.

Finally, I would like to thank my mother for her love and support throughout my graduate program, as well as my Aunty Ngozi, Aunty Uzoamaka and Uncle Chiedu, and other family and friends. I am grateful to have such a strong support network.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Assessments are an important part of the learning process. However, grading student work and providing feedback consumes a lot of instructor time. To address this issue, instructors at the University of Illinois are increasingly turning to computer-based tests, for which student work can generally be graded automatically and correct solutions provided instantly. Computer-based tests are particularly useful for introductory and lower level CS courses, because they allow students to test and debug their code more easily; and grading student code by hand is particularly inefficient.

Exams can be administered asynchronously. Students can take tests at the most convenient time for themselves within a period specified by the instructor. Given that not everyone takes the test at the same time, students may communicate with each other to share questions or solutions. To mitigate this issue, professors often provide different questions for students [1]; therefore, two students seating for the same exam do not necessarily have the same questions. This may lead to concerns about fairness or equity [2, 3, 4]. Professors carefully consider variants [4] in order to address this issue, but little research has been done to determine students' perceptions of fairness for these computer exams.

In this work, we will investigate students' perceptions of fairness of computerized examinations along two separate dimensions: exam security and variance in exam difficulty. We will also consider how pre-exam information given by an instructor affects students' perceptions of fairness. "Pre-exam information" in this context refers to a set of topics to be covered on an exam, along with some general information about the possible variants of questions for an exam. We provide a detailed explanation of the pre-exam information when we explore the dataset.

Exam security influences perceptions of fairness because cheating and other unethical behavior of some students devalues hard work and demoralizes honest students, as well as prevents instructors from gaining an accurate assessment of students' mastery of material that can be used to re-adjust teaching strategies and expectations.

We hope to generate insights to help inform instructors' pedagogical decisions. Our work is also potentially beneficial to other universities who intend to support computer-based testing in the future.

CHAPTER 2: PREVIOUS WORK

Fairness is a cornerstone of ethical teaching [5]. Fairness can broadly be described as a perception based on interpretations of a person's behavior, which subsumes their intentions [5, 6]. The definition of fairness has evolved in the literature. Previously, research on fairness was concerned with issues of distributive justice [5, 6, 7]; perceived fairness of outcomes of an exchange. A fair exchange in an educational context is defined as one in which the ratio of a student's inputs to outcomes equals the ratio of inputs to outcomes of others [6]. More recently, in the literature, "fairness" has been classified into several groups as it relates to coursework: interactional, procedural and outcome fairness [6, 7, 8]. Interactional fairness relates to impartiality and transparency about the policies of a class. Procedural fairness relates to aspects of the course such as the workload assigned, similarity of assessment material to content covered in classes and providing feedback [5]. Outcomes fairness relates to the desire of students that course grades accurately reflect performance: this encompasses the use of accurate assessment instruments and making multiple assessments so that students have several opportunities to demonstrate knowledge [6]. Interactional fairness is described as most important by students, then procedural fairness, then outcomes fairness.

Making exams fair and getting students to believe that exams are fair are challenging tasks. Grading fairness is an important component of pedagogy. With the proliferation of computer-based testing with multiple question variants, there is a growing need to ensure that students are being treated equitably and believe that this is so. One study showed that students became irate and resistant to instruction when they perceived unfairness [9]. Previous work has investigated some factors that form the basis of perception of grading fairness for traditional paper exams: teaching practices that helped students prepare for exams as well as test scoring manipulations, such as curving [8]. In one study, 600 college students at the University of Bradford were surveyed about grading fairness [10]. They were asked to provide data on a number of issues, including whether grades awarded were commensurate with what had been learned in a class, as well as consistency of grades, i.e. if the same procedure was used to assign grades for all students in a class. They found that students at all performance levels were less satisfied with assessment outcomes when they were concerned about grading fairness [5, 7, 9]. A perception about the absence of grading fairness could diminish students' views about the ethics of professors and negatively impact motivation to learn [11]. Simple measures such as curving exams or reducing cutoffs for letter grades do not remedy issues of unfairness, according to students [8]. In this work, we are mostly interested in procedural and outcomes fairness: how do students perceive fairness

of computer-based exams.

Developing tests of equal difficulty and convincing students that the tests are similar are two separate tasks. First, we will discuss some measures professors have taken to determine exam equivalency, and then investigate students' perceptions.

As stated previously, there is a need to convince students of impartiality, which is a higher bar than simply taking actions to ensure equity. To compare question variants for equity, previous work has examined end outcomes (exam scores) and effort [12]. Students should have similar scores if the question variants are equivalent. Also, the effort involved in answering the questions should be the same, including the number of attempts and the amount of time spent per question [12].

Students' perceptions of exam fairness may be affected more than just equivalency of questions asked. For instance, in previous studies [10, 13, 14], students stated that timely feedback from e-assessments helped with learning and impacted their view of fairness. Therefore, a well designed test can still be judged as unfair, if students don't receive information about why they lost points. We will review students' comments in detail to identify other factors that influence their beliefs about fairness.

Previous research has been carried out to determine what instructors think, but very little has been done to determine what students believe. Our work expands on previous studies by examining free-form student comments about various components of computer-based exams.

CHAPTER 3: DATA

We conducted our study at the University of Illinois at Urbana Champaign. This university makes use of a Computer Based Testing Facility (CBTF), a lab on the campus where students can take proctored tests. Instructors develop the assessment, but proctoring is provided by a dedicated staff, which frees up instructors to work on other components of a course. The lab has limited seating capacity (a cap of 80 people). The lab allows for students to take exams asynchronously, with each instructor specifying a time window during which a particular assessment may be completed. A unique exam is generated on the fly for each student, using an algorithm that randomly assigns the student one of the variants that exist for a question. This is done for all exam questions which have variants. Exams are generally graded interactively as each question is submitted and students receive their scores before leaving the lab.

For this study, we investigated perceptions of fairness after a programming exam for a large CS2[1] course: Introduction to Data Structures and Algorithms. This was the 4th exam of the semester out of 6 exams given. The exam was administered during the 10th week of class. The exam had 2 programming questions and a code reading question. A code reading question is a question for which students are given a blurb of code and asked to analyze it and determine its functionality at a high level, without being able to run the code.

The course we collected data on had 453 students. About a week after all students had completed a midterm programming exam, they were asked to fill out a survey, which asked about their experience with the exam in question. However, students were also free to comment on their experiences with previous exams for the same course. They were asked about their thoughts on exam security, exam difficulty variance (i.e. exam equity), helpfulness of pre-exam information provided, general feelings about the code reading question, and what factors led them not to meet their goals, for the students who specified that they did not receive their desired score. A small amount of extra credit was provided as an incentive to complete the survey.

The survey was filled out by 335 out of 453 students, a response rate of 74%. One response was discarded: the student filled out the survey though records indicated that the student did not actually take the exam.

[1] CS2 courses are lower to intermediate level CS courses taken after the introductory courses (so-called CS1 courses) at universities

3.1 SURVEY QUESTIONS

The survey had multiple items. After students completed the survey, we determined that their responses to 3 specific questions would be the most relevant for investigating perceptions of fairness (i.e., not all questions were analyzed for determining perceptions of fairness). We present additional information on the pertinent questions here.

Questions

Exam Security: Because our exams are asynchronous (different people can take them at different times), we have multiple versions of every question so that one person can't trivially tell other people the answers to the exam questions. How important to you is it that we take steps to prevent cheating?

Pre-Exam Utility: Exams necessarily only cover a fraction of the material covered in a course, and one source of variance in student exam scores results from how closely the material studied by the student matches the material on the exam. We sought to reduce this variance by being transparent about what the exam questions would be. How valuable did you find the pre-exam descriptions of the exam questions?

Exam Difficulty Variance: While we do our best to ensure that the exams are of very similar difficulty, it is impossible to make all exams have identical difficulty. In light of your (above expressed) desire for exam security, what would you like to share with us related to variations in exam difficulty?

3.2 PRE-EXAM INFORMATION

The instructor for the course provided information to students about the format of the exam and the content before the students took the test. The instructor informed the students that 3 questions would be on their tests: 2 programming questions and 1 code reading question. The instructor proceeded to show students a specific example of a code reading question. The instructor then explained that the first programming question had several variants and would either ask them to implement one of two data structures or an algorithm for a data structure. The students would not be able to choose which specific variant they received. The same was done for the second question on the exam.

CHAPTER 4: METHODS AND RESULTS

To measure student performance, we used total points earned after the course was completed. The bulk of the points on the midterm programming exam were from two programming questions. As the score distributions for these questions were somewhat binary in nature, the spread from that exam was not large enough to discern nuances in ability. Since students were not restricted to giving feedback about the specific programming exam in question, we believe that their comments will most likely cover their experiences with other tests for the course as well; this assumption was supported by the data upon review. Students had taken 2 programming exams by the time they completed the survey. They had also taken 2 theory exams, for a total of 4 exams. Total points earned in the class at the conclusion of the semester were out of 1000. Ignoring extra credit, the range of scores for the students who completed the survey was 242 to 991, with an average score of 849.9, and a standard deviation of 115.2. We should note that though a large fraction of students responded, we were still concerned about non-response bias. 26% of the class forfeited extra credit that would have been awarded had they completed their survey; this group may be systematically different from students who filled out the survey. We compared the group of students who filled out the survey against those who didn't to see if there was a substantial difference in performance in the class. The group that didn't fill out the survey had a mean score of 722.3, while the group that filled out the survey had a mean score of 849.94. This difference was statistically significant at the 0.01 level, which indicates that there is difference between the two groups of students. It is possible that these students also have different perceptions of fairness, which could impact our results.

4.1 GROUNDED THEORY

To the best of our knowledge, our work is the first to look at detailed student comments about exam difficulty variance for computer-based exams. Here, we used grounded theory to generate insights. Grounded theory is a qualitative research method employed when little is known about what people do and how they think in a given context [15]. No theory should be formed prior to data collection. Data which is gathered should be analyzed with rigorous coding schemes to protect against researchers' biases. Only codes that all researchers agree on should be retained. In our study, two researchers examined the responses to 3 questions on pre-exam helpfulness, exam security and exam difficulty variance, and determined codes that described the comments. The coding was initially done independently, then reconciled

for each piece of student feedback. This was done iteratively for each question. Each student comment could be assigned zero to many codes based on what the student discussed. After investigating the three questions, we identified 81 codes that encompassed different thoughts raised by students, which were later used for analysis. Our codes includes tags for some sentiments that appeared sparingly in our dataset, but may be more prevalent in larger courses. We present the codes, along with descriptions and examples of student responses that match those codes here.

4.2 CODES

ES: Exam Secure. The student was not aware of any cheating going on in the exam. There were 37 instances of this code, where each instance corresponds to a student's comments being assigned this code.
Sample student feedback:
"I think you do a good job at preventing cheating"
"I think the exams are already fairly secure, it would be tough to relay a solution because the questions are relatively complex"

EF: Exam is Fair. The student believed that the exam was generally fair. Students also indicated that they would still consider exams with multiple question variants fair, as long as the difficulty variance was kept low. There were 102 instances of this code.
Sample student feedback:
"Seems pretty fair"
"I feel like the variations in exam difficulty are reasonable as the material presented was previously declared and taught and the time allotted for the exams was more than sufficient."

UNFAIR: Exam is Unfair. The student said that the exam was unfair. There were 27 instances of this code
Sample student feedback:
"I thought I was very prepared for Programming Exam B, I think it's unfair that some people got way easier questions than I did though, because I used my full 2 hours only to find out that some people got a mirror question etc. which are easier codes than what I studied so I could have easily done them. I know there a multiple questions so people don't cheat but I think its unfair when people get two hard ones or two easy ones."

CURVE: Students wanted some kind of curve or other action to balance difficulty of exams. There were 16 instances of this code.
Sample student feedback:
"I feel like each of the programming questions should be curved so they each have about the same average of percent. Let's say student 1 has a mirror question and student 2 has a implement stack question, the average for mirror and stack should be the same."

DQPC: Different Questions Prevent Cheating. Students indicated that giving different questions reduced or eliminated instances of cheating. There were 24 instances of this code.
Sample student feedback:
"I agree with having different versions, but they MUST be equally challenging. Programming Exam 1 was definitely not done this way, and I felt cheated by the course staff. A tree insert is not the same difficulty as a remove in my opinion. There are so many factors to consider for each problem, but in general if students can all agree that one problem is harder, you should be able to see that too."
"I can't imagine that there is too much a change in difficulty but I think it is a small price to pay for more exam security."

DQNN: Different Questions Not Needed. There is no need for multiple variants from the same base question. There were 3 instances of this code.
Sample student feedback: "Cheating in a programming exam seems basically impossible to me since we were given the exact problem descriptions in lecture before. Realistically its not as if someone is going to memorize every single line of code they wrote during the exam, rewrite that for someone else, then have that next person memorize every line as well. Since we were given the problem descriptions beforehand, the most people could do is talk about the high level psuedo code to solve the problem."

DQUF: Different Questions Inherently Unfair. The students who noted this objected to questions with multiple variants as a matter of principle. There were 9 instances of this code.
Sample student feedback:
"Should keep same tests so some people that have the harder version won't do worse than people with the easier version."
"identical exams every year"

DQ→STUDY MORE: Different Questions Implies Students will Study More. Having vari-

ants encouraged people to study more. There were 2 instances of this code.
Sample student feedback:
"I think variety is fine as long as there isn't too much variety. With a little variety, students need to have a greater understanding of the course material. However, too much variety can just be overwhelming."

ASYNC+: Students like asynchronous exams, even if this leads to some unfairness. There were 2 instances of this code.
Sample student feedback:
"I feel that any negative effects from getting a more difficult exam are far outweighed by the benefits of being able to take the exam at CBTF on a computer instead of on paper, and at a time that's convenient for me."

NARROW: Student suggested reducing the period of time during which people could take an exam. There was 1 instance of this code.
Sample student feedback:
"Just change the key data in the problem and narrow the exam time period."

PEPC: Pre-exam information Prevents Cheating. Giving questions before the exam prevents cheating. There were 9 instances of this code.
Sample student feedback:
"I think if we're given pre-exam descriptions, exam difficulty should be identical because we've been given a fair amount of time to study for it."

PEMF: Pre-exam information Makes Fair. Giving questions before the exam makes it fair for everyone. There were 23 instances of this code.
Sample student feedback:
"I do think telling us the topics lets us know how hard it is. So I do not think one would be easier than the other if we knew the topics beforehand. If they were not given, maybe I would say it may be possible for one to be easier than the others."

PE→ C: Pre-exam information leads to Cheating. Providing students with questions before the exam facilitates undesirable exam strategies, such as attempting to memorize code. There were 5 instances of this code.
Sample student feedback:
"Just wanted to say that I could've just gone to geeksforgeeks and memorized code for each

of the questions which I didn't like, because I know that other people would do that."

PELH: Pre-exam information is Less Helpful. The students say that the pre-exam information is less helpful as variation in the actual questions asked on the exam increases (i.e. more variance on questions negates pre-exam utility). There was 1 instance of this code.
Sample student feedback:" [The pre-exam information] was really helpful for the first question but with the iterative question it was harder since there can be variations in implementation."

PEME: Pre-Exam Info Makes Exam Easy. The pre-exam makes the exam much more tractable for students to complete. There were 2 instances of this code.
Sample student feedback:"If the topic is the same, the problems are always similar. So if given pre-exam descriptions, it's better for student to get better grades."

PTS: Question Pool Too Small. Students are contending that the pool of variants for a given question is too low. There were 2 instances of this code.
Sample student feedback: "I heard some 'mutual help' for exams, so how about more versions?"

MSD: More Substantial Differences. Students want more than superficial differences among the variants of a question. There were 4 instances of this code.
Sample student feedback:
"Actually, I think even though questions are different, but they have the same type. It would be more fair if we have different types of problems with the same difficulty."

MORESIMILAR: The questions' variants should be more similar to reduce difficulty variance. There were 14 instances of this code.
Sample student feedback: "The difficulty can varies. For the easy question, I believe different version does not result in huge changes. Since I don't know about the more difficult one and it seem most of people I know in this class did not do a good job on the last programming question. I would say keep the question very similar for the most difficult programming question can reduce the discrepancy across exam versions."

NS: Not Secure. The student felt that the exam or the environment in which it was administered was not secure; it did not meet a student's expectations. There were 5 instances of this code.

Sample student feedback: "I've had classes go to great lengths to try and prevent cheating in the CBTF and it never works"

NI: Needs Improvement. Student indicated that the exam security should be worked on. There were 2 instances of this code.
Sample student feedback: "[Exam security] still needs improvement"

PP: Prefers Paper Exam. The student has indicated a preference for either synchronous exams or for all students to get the same questions. There were 14 instances of this code.
Sample student feedback: "I think 1 paper exam is a good solution for exam security and variation in scores. I really have no idea if there was that much difference in how hard each exam was, this may be over exaggeration from students."

COMM: Communication. The student indicated that he/she (or their classmates) discussed the exam questions with other students, after at least one student had taken it. There were 23 instances of this code.
Sample student feedback:
"A lot of Chinese international students have WeChat groups where they use to discuss answers, I think CS225 staff should look into that."
"I feel that the exams get easier as more information gets out about them. No clue how to account for that, but taking it on Saturday is much easier than taking it on Thursday"

NC: No Communication. The student indicated that he/she didn't discuss the exam questions with other students. There were 12 instances of this code.
Sample student feedback:
"I feel like my exam was sufficiently difficult and cannot speak to others' exams"
"I only took 1 exam, I can't speak on the difficulty variation..."

VB: Variance is Bad. This tag is used to indicate that difficulty variance for questions is bad, as a matter of principle. There were 30 instances of this code.
Sample student feedback: "I think it would be better to have the same difficulty, because even if it isn't randomized, some of us will always get the hardest version."

COMPLEX: The complexity of questions prevents cheating; cheaters can't remember whole blocks of code for a programming problem at a time. There were 5 instances of this code.
Sample student feedback: "I think the exams are already fairly secure, it would be tough to

relay a solution because the questions are relatively complex"

STUDYALL: Pre-exam information encourages studying all the content, or it is still important to study all the content, irrespective of the specific concepts highlighted by the pre-exam. There were 7 instances of this code.
Sample student feedback: "I didn't worry too much about the descriptions as I felt they would've made me tunnel vision"

WORKMORE: Providing pre-exam information makes students work harder and study more. There were 3 instances of this code.
Sample student feedback:"I feel providing different types of questions/topics that CAN be covered on the exam (what you did for programming exam B) is a great solution. I believe this limits cheating and encourages students to work through the material before the exam. It is impossible to give exams of identical difficulties, but I believe this is the best way to give everyone a fair shot at the exam while keeping exam security high."

DIST: Knowing distribution of previous students' performance on an exam is helpful. In addition, getting information about the difficulty of an exam will aid students in determining how long they should study for. There were 4 instances of this code.
Sample student feedback: "A little bit of more insight in terms of how difficult an exam is going to be compared to the previous one would be a good gauge in terms of difficulty"

CPL: Cheating Prevents Learning. There were 2 instances of this code.
Sample student feedback: "Cheating isn't right, but if people do it, they are only harming themselves for interviews in the future. And besides, most people can't remember whole blocks of code, but they might remember just the question, so if anything, the cheater would likely only have an idea of what's on the exam, not the actual solution itself. This exam wasn't multiple choice so you can't simply memorize the answers. Cheating isn't right, but those are my thoughts."

PIH: Perfection Is Hard. The student has indicated that it is difficult for instructors to design tests with multiple variants which all have equal difficulty. There were 25 instances of this code.
Sample student feedback: "I'm not personally concerned with variation in exam difficulty. It's an unfortunate side effect that some people will get easier or harder exams, but that doesn't change the fact that, even if you get the 'hardest' exam version, its still all stuff that

was expect that a student in the class should know (Assuming the cs225 course staff only includes problems that they said would be on the exam, i.e. in the "Topics Covered" page for each exam)..."

DOBETTER: Student indicates some frustration with the current state of exams and is requesting corrective action to make the exams fair. There were 16 instances of this code.
Sample student feedback: "I agree with having different versions, but they MUST be equally challenging. Programming Exam 1 was definitely not done this way, and I felt cheated by the course staff. A tree insert is not the same difficulty as a remove in my opinion. There are so many factors to consider for each problem, but in general if students can all agree that one problem is harder, you should be able to see that too."

GS: Pre-exam information Guided Study. Giving pre-exam information helped focus study efforts of students. There were 62 instances of this code.
Sample student feedback:
"They were very useful in that it helped me narrow down what code I would need to write, but this last time I got a little confused because the wording wasn't as clear? I also felt like the second part of the exam didn't completely reflect what I was supposed to study because of that."
"I studied what the pre-exam description told me to and I still didnt do well so maybe a little more detail would be appreciated"

RA: Pre-exam Reduced Anxiety. Giving pre-exam information led to a reduction in the anxiety of a student before he/she took an exam. There were 12 instances of this code.
Sample student feedback: "The pre-exam descriptions were exceedingly helpful. At the very least the drastically reduced my anxiety going into the exam."

HELP: Pre-exam information was Helpful. Student indicated that the pre-exam was beneficial in some manner. There were 74 instances of this code.
Sample student feedback: "They were helpful. I liked having an idea about what sort of questions would be on the exam to study. I tried to generally study all the material and not anything specifically, so this didn't affect me on the questions."

MISLEADING: Pre-exam information was Misleading. The pre-exam may have emphasized course content that was not present on the exam. There were 10 instances of this code.
Sample student feedback:

"They gave me a set expectation for the exam, which made it harder to get around the varied problem."
"There are things tested that are not described in the descriptions."

NOT USEFUL: The pre-exam information was not useful in guiding study before the exam. There were 8 instances of this code.
Sample student feedback: "The pre-exam description was very vague, not very useful, especially when the exam has very different topics"

NO IMPACT: The pre-exam information was appreciated but had no impact on either studying or performance on the test. There were 3 instances of this code.
Sample student feedback: "Had no affect on preparation other than slightly more focused, but the big impact was going into the exam with confidence rather than seeing something and being discouraged if you don't immediately recognize the type of problem"
"Didnt really change anything, was going to study them anyway"

MORE: The student indicated that he/she wanted more resources or information to prepare for the exam, or the provided information was unclear. There were 44 instances of this code.
Sample student feedback:
"These descriptions helped me narrow down what to study and really familiarize myself with, however, it did not prepare me for the variances in questions and was a bit too vague."
"I studied what the pre-exam description told me to and I still didnt do well so maybe a little more detail would be appreciated"

PRACTICE: This is a subcategory of the MORE code; student wanted more resources, specifically more practice problems or exams to help with preparing for the main test. There were 12 instances of this code.
Sample student feedback: "I hope that we can also have practice exam for programming exam."

EARLIER: The student wanted the pre-exam information to be disseminated by the instructor earlier. There were 2 instances of this code.
Sample student feedback: "I wish we were given these descriptions early because they were only given out a couple days before the exam, which means there may not be a lot of time to review those topics, and the descriptions would then be as useful as if they were just seen on the exam. I think having some vague idea of what to expect on each question is very impor-

tant to have and I really hope to see it again for the last programming exam and final exam! (P.S. giving a study guide like this beforehand actually makes students study the content even more, which makes us even more prepared for interviews and other work in the future.)"

EARLIER → EF: Pre-exam info Earlier implies Exam Fair. Releasing the pre-exam information early makes the exam fair. There was 1 instance of this code.
Sample student feedback: "I feel like if we have enough and equal time to review and understand the different concepts then it would be equal and there wouldnt be too many issues"

ENOUGHTIME: The student indicate that the pre-exam information was provided early enough. There was 1 instance of this code.
Sample student feedback: "I feel like if we have enough and equal time to review and understand the different concepts then [fairness] would be equal and there wouldn't be too many issues"

UNWANTED: The student suggests potential problems with providing pre-exam information, or wishes that the pre-exam information was not available at all. There were 8 instances of this code.
Sample student feedback: "I was able to focus my practice. But I also feel I lost out on studying all the material covered, had you not mentioned the questions."

HARD: The student indicated that the exam was hard or wanted it to be easier. There were 19 instances of this code.
Sample student feedback:
"I feel like my exam was harder than I expected, even though I studied all the practice questions."
"The exam can be easier than now."

NOT HARD: The student indicated that the exam was not hard. There were 3 instances of this code.
Sample student feedback: "I don't really know how other exams
compared to mine, but my iterator for a reverse level traversal was not too difficult."

VERSION ANX: Student was concerned that he/she might get a harder version from the pool of variants; or the student indicated that he/she got a harder version than other students in the class after the exam. There were 4 instances of this code.

Sample student feedback:
"I think it was useful, but I just freaked out when I got the harder first question, which I hadn't studied for, so it took longer to get."
"I think each of the questions should be of very similar difficulty. It feels really bad to get the frustrating to get the 'bad' exam question."

POST EXAM REVIEW: Students wanted to review the exam after they had completed the assessment. There were 2 instances of this code.
Sample student feedback: "I don't know much about the exam security, but I definitely heard about people say how easy the exams were when I found them difficult. I think one solution might be to invite students to review different versions of exams and discuss their difficulty at the end of this semester. This action should at least help future students in maintaining a fairer scale of exams. "

MORE SMALLER: Student posited that having more questions with smaller components or fewer tasks to accomplish per question helped to reduce variance. There were 4 instances of this code.
Sample student feedback: "To make the variations in exam difficulty lesser it would be better to have more questions on the exams and the difficulty of the tests could get closer to the average."

DEBUG: The student explained that more resources needed to facilitate debugging on exams; writing code without access to popular code discussion forums was not ideal. There were 6 instances of this code.
Sample student feedback: "I think the hard part about the exam is to debug without the help of google. It is not really on if you know how to implement more on how well you know the library functions."

UNDERSTUDIED: The student conceded that he/she did not study enough for the exam. There were 11 instances of this code.
Sample student feedback: "I did study the specific algorithms for insertion which helped expedite that process a lot, but when given the specifics with a const iterator, I struggled to work with the insertion. I think I should have studied more on the iterator, but the only real in depth iterator we implemented was on the MP, and that was a fair bit more complex than what was on the test. Even looking at resources for iterators, I struggled to understand them to the level that was necessary for the exam."

LEARNING: Student emphasized the importance of learning the material covered in class. There were 2 instances of this code.
Sample student feedback: "I think it is fair exams are randomized to an extant because it requires students to study over a broader range of subjects. This should only benefit the student who is looking to learn more about data structures."

DEPTH: Students indicated that they used their allotted study time to study pre-exam topics in more depth. There were 7 instances of this code.
Sample student feedback: "I found the pre-exam descriptions very helpful because they helped me focus my studying on specific topics and making sure I completely understood those prior to the exam."

EFFICIENT: The student indicated that using the pre-exam information, they avoided studying topics which would not be covered on the exam. There were 5 instances of this code.
Sample student feedback:
"Thank you for providing us with the questions, it made studying for this test a much more time efficient and less stressful process."

NO PROCRASTINATE: Students indicated that providing the pre-exam information reduced their tendency to procrastinate preparing for exams. There was 1 instance of this code.
Sample student feedback: "The pre-exam descriptions were extremely useful. Without these sorts of descriptions, there is a greater level of uncertainty which results in more time studying irrelevant topics and probably worse outcomes. Knowing the exam questions in advance gave me a lot of confidence and stopped me from putting the exam off. I signed up for one of the earliest time slots and did very well thanks to the pre-exam descriptions."

EXAMDIFFHW: Relates to the exam being different from the material covered in homework and labs. There were 16 instances of this code.
Sample student feedback:
"Exams should have a similar distribution of topics as homeworks and lectures. When things are barely covered in either of those, I think it's not something that belongs in an exam, and that's a factor that can make exams much more difficult than they should be."
"For the iterator part, I think it was the same way as lab, but in the exam is more focus on

the basic operator, but for lab we actually pay more attention to [different] algorithm other than basic operator."

MEMORIZED: Student indicated that he/she attempted to memorize blurbs of code as a study strategy, but this did not improve performance on the exam. There were 2 instances of this code.
Sample student feedback: "As I said above, I studied all of the possible problems rigorously. However, the implementations for all of the problems were completely different than the ones we did in the labs and MPs, as well as the "typical" implementation online. Therefore, all of my studying was mostly useless, because it came down to how long I would take to figure out how to work around the random implementation."

VARTOOBIG: Student said that the difficulty varied substantially for the question variants. There were 62 instances of this code.
Sample student feedback:
"I feel like the queue questions was by far easier than the other two, which is what I got and was happy about."
"The exam version matters because [one] can easily be understood and the other not."

CENTRAL LIMIT: Student posited that the number of easy and hard questions students from the list of possible variants tends to balance out. There was 1 instance of this code.
Sample student feedback:"I think it might be unfair to vary exam difficulties because not every student got the same level of difficulty on the exam. I know CS 173 also has many exam variations of different difficulties. However, I think that their system is fair because there are a lot more examinations in that course than in 225, so the randomness of difficulty will balance out. In 225's case there's only 3 programming exams and having students get harder problems on the exam could significantly hurt their grade because of the weight on the exam"

PREEXAMVAR: The student believed that the difficulty variance of the exam would be unacceptably large based on the pre-exam information. There were 2 instances of this code.
Sample student feedback: "I actually felt like the questions (based on the possibilites announced beforehand) had very different difficulty levels."

MORETIME: Student indicated that more time was needed to actually complete the test. There were 4 instances of this code.
Sample student feedback: "As long as the instructor feels they have prepared their students

well enough to answer any of the exam questions, regardless of the difficulty, the student should be able to answer. The problem may be time given at that point. If a problem is more difficult the student should be given more time."

CHOICE: Indicated that it would be preferable for students to have a choice of which questions to solve on the exam. There were 6 instances of this code.
Sample student feedback: "I think it would be better to either have simpler but more questions testing more of the material covered and/or have more questions but the student only having to choose to answer a fraction of the question. So for example, if there are 5 questions then maybe, the student has to only answer 3, that way students are going for questions that cover material they understand more."

EXAM UNCLEAR: Student stated that the exam itself was confusing. There were 19 instances of this code. There were 3 instances of this code.
Sample student feedback: "I felt very confused because the exam prompt did not give explanation to how some functions were supposed to work or how the iterator was supposed to work. I had no clue what the ++ operator was doing because i wasnt sure what its intended use was. Its hard to work on a structure you get thrown into in a time crunched situation with no explanations"

PEOPLEVAR: Variation in performance or perceptions of difficulty was due, in part, to knowledge variance across students. A particular student may have better familiarity with some of the concepts covered on the exam than other components of the course. There were 11 instances of this code.
Sample student feedback: "Some students might have problems that they simply understand better and do quite well, despite not knowing how to do the other possibilities at all."

FAIRGAME: Student indicated that it was completely acceptable to ask any questions about any covered material, because students should have learned all the content covered in class. There were 9 instances of this code.
Sample student feedback: "I think they are about the right, I wasn't too concerned about the variation in difficulty, because any sane person will probably choose to prepare for all the topics anyway."

I GOT LUCKY: The student stated that the difficulty of the questions varied, and explained that he/she got one of the easier variants. There were 5 instances of this code.

Sample student feedback: “I feel like the queue questions was by far easier than the other two, which is what I got and was happy about”

UNLUCKY: The student believes they got a hard variant of a question. There were 6 instances of this code.
Sample student feedback: “I thought my question was harder than the others. I know people who got lucky and were able to hard code certain values to get their answer right. I know I did not have that luxury. I felt like my questions were harder.”

GENOK: The student indicated that exam variance wasn’t a significant issue in the previous exams in the course. There was 1 instance of this code.
Sample student feedback: “I think the coding tests so far are pretty fair. I have heard that the variance of difficulty in final would be much larger, so I hope you could manage this.”

PARTIAL: The student wanted more partial credit to be awarded on the exam. There were 4 instances of this code.
Sample student feedback:
“Sample exam just like in theory exam might help lower the variation ; Giving more partial credit maybe helpful”
“I just wish more partial credit was available”

UNDERPREPARED: A student indicated that other students complained because they didn’t study enough. There was 1 instance of this code.
Sample student feedback: “I think your method for this exam is good enough. People complain about this exam because they did not study hard enough.”

COVERAGE: Students stated that the exam material was not sufficiently covered in either the lecture, homework, labs or other course components. There were 3 instances of this code.
Sample student feedback:
“Do not give problem that rarely talked about in lessons”
“Exams should have a similar distribution of topics as homeworks and lectures. When things are barely covered in either of those, I think it’s not something that belongs in an exam, and that’s a factor that can make exams much more difficult than they should be”

RELEASESTATS: The student wanted the average score for each variant to be released. There were 3 instances of this code.

Sample student feedback: "Release the score deviation averages median and mod for different versions so that it would be more clear for the students whether the questions were equally hard or easy."

HQL: Student discussed the possibility that exams taken later in a test window could be purposefully designed by instructors to be harder than exams taken earlier in that window. This code was assigned to students who were either in favor of or against harder questions based on when an exam was taken. There were 5 instances of this code.
Sample student feedback:
"I feel that the exams get easier as more information gets out about them. No clue how to account for that, but taking it on Saturday is much easier than taking it on Thursday"
"When I talked with many of my friends after the exam, I found that the exam questions on the Friday for Programming Exam II seem to be more tricky than the exams in the previous days, due to these unusual limitations all together."

CHEATERS WILL WIN: Student argued that cheaters would find a way to circumvent any security measures put in place for the exam. There was 1 instance of this code.
Sample student feedback: "Though the multiple questions makes sense in terms of exam security, it does feel terrible to hear your friend had a question that you know you could've implemented in a heartbeat. I understand that there is not much that could be changed to keep the prevention of cheating, but I feel as though if people really wanted to cheat, they could find a way. In a perfect world, I wish the exams were all the same, but I understand why it is done the way it is."

ES != ED: Exam Security is not equal to Exam Difficulty. The student doesn't believe that there is a relationship between exam security and exam difficulty variance. There was 1 instance of this code.
Sample student feedback: "I don' really see exam security and exam difficulty as related. If there are multiple possible options for a particular question, then each should be roughly equivalent in terms of difficulty."

OBJECTIVE: The student thinks the question has an objective difficulty that is obvious to everyone. There was 1 instance of this code.
Sample student feedback: "Some algorithms are objectively more difficult to code. Clear was harder than implementing queue or stack."

FLEXIBILITY: The student complained about the ability to change some of the code provided for a question. There were 2 instances of this code.
Sample student feedback: “The difficulty of a problem not only depends on the problems itself but also how many ”freedom” we had. For example, I got the mirror problem in the exam. Although it is the same problem as in the lab but we cannot have helper function (I really dont know how to declare a function in cpp file since we never did that). So this became a relatively hard question for me.”

RETAKE: The student wants a so-called second chance exam, a retake. There was 1 instance of this code.
Sample student feedback: “I hope we can retake the exam if the first try is not enough, for like 80% sth.”

NA: Not Applicable. This tag represented content that diverged from topics of interest. It was also assigned to comments where people explained they had nothing to say. This code was not used for further analysis as it did not contribute to our understanding of students' perceptions; no count of the number of instances is reported.

After we tagged the students' responses with codes, we computed the inter-rater agreement. This is a metric to determine the level of agreement among the researchers and provide information regarding the reliability of the codes. For our calculation, we computed the total number of instances of agreement and then divided this by the total number of instances of agreement and disagreement.

In Table 4.1, we computed an interrater reliability score for each of the three survey questions.

4.3 DISTRIBUTION OF CODES

For reference, we have included the distribution of codes, from the most to least frequent codes in Figure 4.1. Some codes occurred sparingly in the data, but captured important facets of exams (e.g. MEMORIZED or FLEXIBILITY).

The specific counts for the most common codes are included in Table 4.2. From Table 4.2, we can see that the “exam fair” tag was the most popular code. It was listed by 102 students out of 334 students, which translates to about 30.5% of the class actively describing the exam as fair.

4.4 ANALYSIS OF EXAM DIFFICULTY

To determine if there was any difference in performance among students who expressed different sentiments about the assessment, we collated the tags into groups of "easy", "hard" or "neutral." This also addressed the data sparsity problem arising from the fact that we have 81 dimensions (i.e. codes). Neutral tags were not coded explicitly. Rather, any tag not explicitly categorized as easy or hard was labeled as neutral.

Tags characterized as reflecting EASY: ASYNC+, PE →C, PEME, PTS, MSD, COMPLEX, STUDYALL, WORKMORE, CPL, NOIMPACT, ENOUGHTIME, UNWANTED, NOT HARD, CENTRAL LIMIT, TIMEOK, FAIRGAME, IGOTLUCKY.

Tags characterized as reflecting HARD: UNFAIR, CURVE, DQNN, DQUF, MORESIMILAR, PP, VB, DOBETTER, HARD, VERSION ANX, UNDERSTUDIED, EXAMDIFFHW, VARTOOBIG, TIME, UNLUCKY, GENOK.

"NEUTRAL" tags can be inferred by picking out the tags not listed in either the hard or easy categories.

After tags were categorized into groups, average student performance for each group was computed. There was a significant difference between the "EASY" and "HARD" groups at the 0.01 significance level, which is evidence of performance differences for students who gave different feedback.

4.5 RESPONSE THEMES

Two researchers analyzed the data independently to discover themes in the feedback provided by students. One researcher (investigator 1) grouped themes top-down based on broad topics students identified and discussed, while the other researcher (investigator 2) grouped themes bottom-up, using agglomerative clustering to identify tags that had similarity in meaning or sentiment.

The first investigator identified the following themes:

- Exam Related: MORE, PRACTICE, HARD, TIMEOK
- Transparency Related: RELEASESTATS, DIST
- Variation Is Problematic: UNFAIR, VB, DQUF, I GOT LUCKY, UNLUCKY
- Students suggestions for improvement: CURVE, CHOICE, PARTIAL
- Exams are fair: EF, GENOK

- Exams are secure, cheating prevented: ES, DQPC, COMPLEX, MEMORIZED
- Not secure enough: NS, Needs Improvement (NI), PTS
- Learning related: STUDYALL, LEARNING
- Pre-exam good: PEMF, GS, RA
- Pre-exam negative affect: MISLEADING, NOT USEFUL
- Pre-exam neutral: PELH, NO IMPACT
- People's failure own fault: UNDERSTUDIED, UNDERPREPARED
- Surprising and contradictory opinions: PEOPLE VAR, DQNN.

The themes that related to utility of pre-exam information were analyzed together; this comparison was the only logically sound one to make, as these themes all dealt with a single dimension of the exam. The results are shown in Table 4.4 and 4.5.

The second investigator identified the following themes by iterative rounds of clustering.

- Maximize knowledge gain: FAIR GAME, LEARNING
- Exam Fair: Exam Fair, Different Questions Prevent Cheating,
- Perfection is Hard
- More Tools Needed: DEBUG, CHOICE
- Pre-exam helpful: GS, RA, Helpful
- Pre-exam makes fair: PEMF, TIME OK
- People Luck: PeopleVar, I GOT LUCKY, NOT HARD
- Traditional preferred: Prefer Paper, Complex, Different
- Questions Not Needed
- Needs major improvement: More, Practice, VarTooBig.
- Unprepared: Understudied, Underprepared, Memorized.

We compared two themes, "exam fair" and "needs major improvement", to determine if there was any difference in performance of students who had different perceptions about the quality of the exam. Average scores for the "exam fair" group were found to be significantly different from the "needs major improvement" groups. We also included the "maximize knowledge gain" theme for comparison because it reveals unexpected results about students' beliefs. Students who were characterized by this theme were particularly eager to learn as much as possible, and contended that exams could include any material covered in the course.

4.5.1 Consolidated Themes

After discussion, we consolidated our findings and generated themes to characterize the most frequently recurring sentiments in the data.

1. Exams are secure; cheating is prevented.

   Sample responses:
   "I do this whole class on my own. I love how secure the exams are, feels like a level playing field, and even if it isn't at least you guys help me feel that it is with your efforts to randomize questions. Also emphasizes studying all the material not just some."
   "I think the exams are already fairly secure, it would be tough to relay a solution because the questions are relatively complex"

2. Randomly assigned questions are perceived to be unfair, either abstractly or in a student's specific situation.

   Sample responses:
   "Should keep same tests so some people that have the harder version won't do worse than people with the easier version."
   "If there is a constant change in the difficulty of the test, then it must be possible for some people [to] always encounter difficult versions. In addition to fairness, then, the test becomes a test of luck."

3. Some students are concerned about cheating. In addition, some students indicated that their classmates communicate with friends about an assessment after taking the exam. Fortunately, the impact of this unethical behavior is mitigated when we have multiple variants of questions [1].

   Sample responses:
   "People do ask others what questions they had"

"Students often share specifics outside of the exams. I'm sure this is widely known already. Sharing the [preexam information] beforehand helps prevent this."

4. Students like the pre-exam information; it reduces anxiety before tests and guides study.

   Sample responses:
   "I'm very thankful for [the pre-exam information] as they greatly helped reduce my stress before the exam."
   "This did take a lot of pressure off of me, and I think that moving forward, this is a great resource. Instead of trying to review everything in a more shallow sense, this allows us to delve into each topic because frequently, a lot of the questions on exams demand that you know more about a topic than what is superficial or rudimentary about it. I would appreciate it if this was kept for future exams."

5. Exams in practice were sufficiently fair.

   Sample responses:
   "Small amount of difficulty variance is reasonable."
   "I feel providing different types of questions/topics that CAN be covered on the exam (what you did for programming exam B) is a great solution. I believe this limits cheating and encourages students to work through the material before the exam. It is impossible to give exams of identical difficulties, but I believe this is the best way to give everyone a fair shot at the exam while keeping exam security high."

6. Instructor intervention is needed to ensure parity; many suggestions of how to improve exam fairness were provided.

   Sample responses:
   "Would it be possible to curve certain questions to match other questions' grade distributions?"
   "If practice questions were made available, I feel like there'd be less complaint about difficulty"

7. More instructor transparency is needed.

   Sample responses:
   "Release the score deviation averages median and mode for different versions so that it would be more clear for the students whether the questions were equally hard or

easy."

"I don't know much about the exam security, but I definitely heard about people say how easy the exams were when I found them difficult. I think one solution might be to invite students to review different versions of exams and discuss their difficulty at the end of this semester. This action should at least help future students in maintaining a fairer scale of exams."

4.6 SENTIMENT DISTRIBUTION BY QUESTION

Here, we investigate the distribution of student feedback by question variant. Previously, we saw evidence of differences in performance when we looked at students who used various tags that we characterized as either reflective of the "hard" or "easy" difficulty groupings. However, to determine if the questions are perceived to be differentially difficult, we must investigate the questions directly and examine if the sentiments are similarly distributed for each question variant. If the distributions are unequal, this could constitute evidence of difference among question variants. We analyzed data for the 2 programming questions on the midterm exam. The first programming question had 6 variants, asking students to either implement a simple data structure or implement a method for a tree. Students were randomly assigned one of six variants listed below.

Question 1

- Implement a queue
- Implement a stack
- Implement cloning for a tree
- Implement find function for a tree
- Implement insertion for a tree
- Implement tree mirroring

The second question on the exam also had 6 variants, asking students to implement a traversal of a tree and list a specific set of elements.

Question 2

- Implement an even preorder traversal
- Implement a negative preorder traversal
- Implement an in-order traversal of nonnull tree elements
- Implement an in-order traversal of odd tree elements

- Implement an in-order traversal of positive tree elements
- Implement a preorder traversal of positive tree elements.

For each of the six variants of question 1, we noted the number of students who got that variant. The same was done for question 2. We also looked at combinations of variants of question 1 and question 2 together. We expect roughly equal numbers of students to receive each question or combination of questions because the question variants were assigned randomly. Following tests for statistical significance, no significant differences between the number of students who received any question or combination of questions was found. This suggests that differences in the number of students who give positive or negative comments for the respective question variants are not due to a particular question having more (or less) students providing feedback than other questions. Rather, a difference in sentiment is more likely due to a specific characteristic of the question that distinguishes it from other questions; any such question is, by definition, unfair.

To compare the distributions, we used the chi square test for independence. The test helps us determine if there is an interaction between question variant and sentiment expressed. For our analysis, we only included the most pertinent codes which characterized exam difficulty variance. Specifically, we looked at the number of students whose comments were characterized by the following tags: EF, PEMF, HELPFUL, DO BETTER, DQUF, EXAMDIFFHW, HARD, MISLEADING, EXAM UNCLEAR, UNFAIR, UNLUCKY, VARTOOBIG. The tags which indicated negative sentiments were consolidated into a "NOT FAIR" tag, because some of the individual codes had low counts that could invalidate the chi-square assumptions.

It is important to note that "NOT FAIR" is a composite tag which includes codes that reference exam characteristics identified as being indicative of unfairness in the literature. The UNFAIR code is just one of several codes collated to create the "NOT FAIR" composite.

"NOT FAIR" codes: UNLUCKY, DOBETTER, DQUF, EXAMDIFFHW, HARD, MISLEADING, EXAM UNCLEAR, UNFAIR, VARTOOBIG.

The first number in each *Codes* cell for Table 4.8 and Table 4.9 represents the observed (i.e. actual) number of times a given code was associated with a question variant. For instance, in Table 4.8, we had 16 instances in the data where students who got the queue_impl variant described the exam with EF or PEMF or both. The second number in each cell represents the expected number of times a given code should be associated with a question variant, under the assumption that there is no relationship between question variant and sentiment expressed. For instance, in Table 4.8, we expected that queue_impl would have approximately

19 instances of EF or PEMF or both.

There was no evidence of a significant difference in feedback based on the question variant received for either question 1 variants or question 2 variants; the p-values obtained after the chi-square test were 0.91 and 0.42 respectively, which are much greater than the 0.05 significance level.

4.7 FIGURE AND TABLES

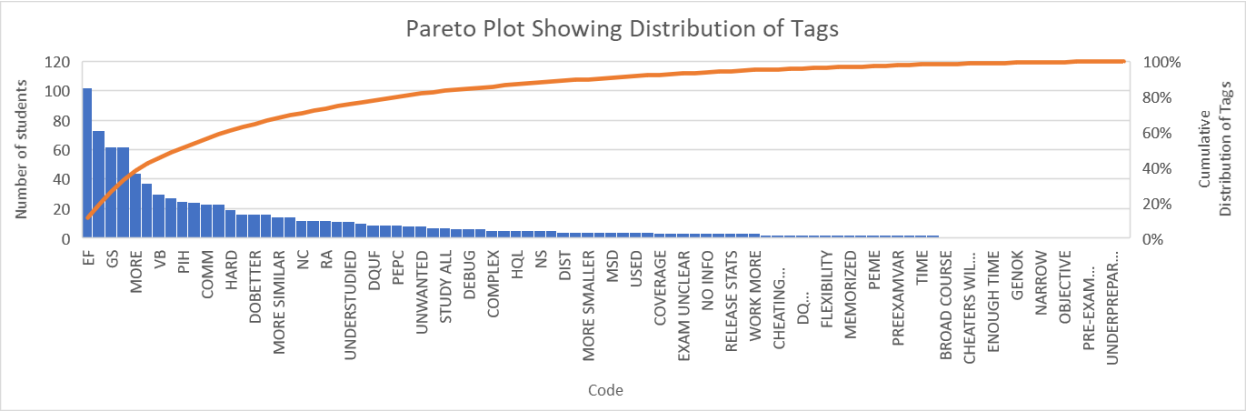

Figure 4.1: Distribution of Codes for Student Responses

Table 4.1: Interrater agreement for coding of survey questions

Exam Dimension	Inter-rater reliability
Difficulty Variance	0.74
Security	0.75
Pre-exam Information	0.73

Table 4.2: Most Common Codes

Code	Frequency
EF	102
HELPFUL	74
GS	62
VARTOOBIG	62
MORE	44
ES	37
VB	30
UNFAIR	27
PIH	25
DQPC	24
COMM	23
PEMF	23
HARD	19
CURVE	16
DOBETTER	16
EXAMDIFFHW	16
MORE SIMILAR	14
PP	14
NC	12
PRACTICE	12

Table 4.3: Two Sample T-Tests

Comparison	Diff	CI	p-value
EASY vs HARD	-39.81	[-70.81, -8.81]	0.0075***
EASY vs NEUTRAL	-19.08	[-47.1774, 9.0174]	0.2483
HARD vs NEUTRAL	20.73	[-5.6082, 47.07]	0.1547

Table 4.4: Average Score for Students in Theme

Theme	Average Score	Standard Deviation
Pre-exam good	864.8	109.5
Pre-exam neutral	925.6	30.9
Pre-exam negative affect	819.2	115.98

Table 4.5: Comparison of Average Performance Based on Pre-Exam Themes

Comparison	P-value
Pre-exam good vs pre-exam neutral	0.0002 ***
Pre-exam good vs pre-exam negative affect	0.0746
Pre-exam negative affect vs pre-exam neutral	0.0003 ***

Table 4.6: Average Performance Based on Themes

Theme	Average Score	Standard Deviation
Maximize knowledge gain	903.8	68.2
Exam Fair	874.3	105.5
Needs major improvement	830.6	123.4

Table 4.7: Comparison of Average Performance Based on Themes

Comparison	P-value
Exam Fair vs Needs major improvement	0.001
Exam Fair vs Maximize knowledge gain	0.1024

Table 4.8: Chi Square Test of Independence for Question 1 Variants

Question 1 Variant	EF + PEMF	HELPFUL	NOT FAIR	TOTAL
queue impl	16 (18.8)	13 (10.96)	18 (17.3)	**47**
stack impl	23 (20.8)	11 (12.1)	18 (19.1)	**52**
treenode clone	19 (19.97)	11 (11.7)	20 (18.4)	**50**
treenode find	19 (16.8)	12 (9.8)	11 (15.3)	**42**
treenode insert	21 (22.4)	14 (13.1)	21 (20.6)	**56**
treenode mirror	27 (26.4)	12 (15.4)	27 (24.3)	**66**
sum	**125**	**73**	**115**	**313**

Table 4.9: Chi Square Test of Independence for Question 2 Variants

Question 2 Variant	EF + PEMF	HELPFUL	NOT FAIR	TOTAL
even_preorder	20 (23.96)	19 (13.99)	21 (22.0)	**60**
negative_preorder	16 (20.8)	8 (9.8)	18 (15.43)	**42**
nonnull_inorder	14 (19.97)	13 (11.2)	20 (17.6)	**48**
odd_inorder	19 (16.8)	10 (9.8)	11 (18.0)	**49**
positive_inorder	25 (22.4)	9 (11.7)	21 (18.37)	**50**
positive_preorder mirror	31 (26.4)	14 (14.9)	27 (24.3)	**64**
sum	**125**	**73**	**115**	**313**

CHAPTER 5: ADDITIONAL EXPLORATION

In this section, we discuss possible techniques for investigating students' perceptions of fairness on a large scale; i.e. for classes with many hundreds or thousands of students. We begin to form a framework with which instructors can gain important information about perceived fairness of their exams and use this information to make decisions.

5.1 DOCUMENT SIMILARITY

Cosine similarity is a ubiquitous method for determining how similar two vectors are. Here, we use it to detect perception differences for exam question variants. If two variants of a question are similar, then we also expect the feedback given for each of those variants to be the same. To test this, we collated the comments for each question variant. We then formed a document for each question variant using the comments associated with that particular variant. The similarity between each possible pair of documents (i.e. question variants for a given question) was computed. We present the similarity matrix for feedback for variants for one programming question in Table 5.1.

The cosine similarity values are relatively high, which may indicate that the documents are similar. This finding is supported by our feedback analysis. In future work, we intend to test the reliability of such an approach by examining a larger dataset. For a more varied dataset, we may need to use word vectors in order to capture semantic similarity, the so-called soft cosine measure[16], for individual terms used to describe fairness. For instance, "bad" and "not good" are essentially equivalent and should be treated as such. If any pair of question variants have a low similarity score, where "low" is defined by an instructor, then we can suggest that the instructor conduct a thorough manual review of the responses for those two question variants. This reduces the amount of work an instructor must perform to gain insight.

Table 5.1: Similarity Matrix of Student Feedback By Question Variant

	PosInorder	PosPreorder	NegPreorder	NonnullInorder	OddInorder	EvenPreorder
PosInorder	1	0.961	0.939	0.948	0.953	0.939
PosPreorder	0.961	1	0.952	0.955	0.956	0.95
NegPreorder	0.939	0.952	1	0.94	0.943	0.93
NonnullInorder	0.948	0.955	0.94	1	0.951	0.937
OddInorder	0.953	0.956	0.943	0.951	1	0.935
EvenPreOrder	0.939	0.95	0.93	0.937	0.935	1

CHAPTER 6: CONCLUSIONS

6.1 SUMMARY

In this thesis, we investigated students' fairness perceptions for exams with multiple question variants. We identified codes and themes that characterized student perceptions and sentiments on programming examinations. We then performed various tests to determine students' perceptions of fairness. We found evidence of a difference in performance among students who described the exam with "easy" tags such as PE→C (pre-exam leads to cheating) as opposed to "hard" tags, such as DQUF (different questions inherently unfair). Further analysis revealed that there was no difference in the sentiment distribution by question variant.

Our results show that students are not significantly more likely to complain about a specific variant, or to give positive feedback about any variant. Therefore, we suspect that students' beliefs about exam fairness were impacted by factors other than the specific questions they received on exams. Perceptions of fairness were based, in part, on performance.

Because there is randomness involved in assigning question variants, a fraction of students may be inclined to contend that their performance was heavily impacted by luck [17] and that an exam is not fair. From previous work [18, 19], we know that students are more likely to attribute unexpected outcomes on an assessment to a variable, external factor . Nonetheless, instructors can engage in attribute retraining [20]: giving feedback to students and suggesting appropriate learning strategies for CS and other courses. Positive instructor intervention may lead students' attributions to change, spurring better performance and satisfaction in courses.

6.2 TOPICAL MINING FOR INVESTIGATING FAIRNESS

For future work, we intend to leverage topical mining to assist in detecting perceptions of fairness. LDA is a generative topical mining model. It allows us to identify a mixture of topics within a document, and the terms that comprise that topic[21]. After we dissect the student feedback data into documents based on some interesting criteria (such as the question variant or when in the testing period an exam was taken), we can identify the most common topics (themes) for each document as well as the most common words used to characterize that topic by running LDA. If the distribution differs significantly among documents, we may recommend that the instructor manually review the feedback to gain

additional insight about students' perceptions.

In our specific case, we had difficulty generating good results with LDA given our limited data; we had only 12 documents with 50-60 sentences per document. This exploration is more feasible for larger classes where students supply more feedback. Therefore, we leave this line of inquiry for future work.

6.3 LIMITATIONS

We performed a qualitative study to investigate students' perceptions of fairness using data for one CS2 course at the University of Illinois; many other courses use computer-based tests, so our results may not be fully generalizable. It is possible that students who take lower level or more advanced classes have different perceptions of fairness.

In addition, the students who did not respond to the survey performed significantly worse in the course than those who did. The non-respondents also forfeited extra credit. It is possible that this group of non-respondents would have given feedback that was systematically different from what we received from the rest of the class.

We used final scores in the course in our analysis, so the relationship between performance and feedback may have been distorted by the extra data (specifically, the other assessments and course assignments that comprised part of the final score for the course).

REFERENCES

[1] B. Chen, M. West, and C. Zilles, "How much randomization is needed to deter collaborative cheating on asynchronous exams?" in *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM, 2018, p. 62.

[2] D. Spinellis, P. Zaharias, and A. Vrechopoulos, "Coping with plagiarism and grading load: Randomized programming assignments and reflective grading," *Computer applications in engineering education*, vol. 15, no. 2, pp. 113–123, 2007.

[3] B. Bridgeman, C. Trapani, and J. Bivens-Tatum, "Comparability of essay question variants," *Assessing Writing*, vol. 16, no. 4, pp. 237–255, 2011.

[4] S. Jordan, H. Jordan, and R. Jordan, "Same but different, but is it fair? an analysis of the use of variants of interactive computer-marked questions," 2011.

[5] B. Whitley, D. V. Perkins, D. W. Balogh, P. Keith-Speigel, and A. F. Wittig, "Fairness in the classroom," *APS Observer*, vol. 13, no. 6, pp. 24–7, 2000.

[6] R. C. Rodabaugh, "Institutional commitment to fairness in college teaching," *New Directions for Teaching and Learning*, vol. 1996, no. 66, pp. 37–45, 1996.

[7] M. B. Houston and L. A. Bettencourt, "But that's not fair! an exploratory study of student perceptions of instructor fairness," *Journal of Marketing Education*, vol. 21, no. 2, pp. 84–96, 1999.

[8] M. E. Gordon and C. H. Fay, "The effects of grading and teaching practices on students' perceptions of grading fairness," *College Teaching*, vol. 58, no. 3, pp. 93–98, 2010.

[9] R. M. Chory-Assad and M. L. Paulsel, "Classroom justice: Student aggression and resistance as reactions to perceived unfairness," *Communication Education*, vol. 53, no. 3, pp. 253–273, 2004.

[10] J. Dermo, "e-assessment and the student learning experience: A survey of student perceptions of e-assessment," *British Journal of Educational Technology*, vol. 40, no. 2, pp. 203–214, 2009.

[11] R. M. Chory-Assad, "Classroom justice: Perceptions of fairness as a predictor of student motivation, learning, and aggression," *Communication Quarterly*, vol. 50, no. 1, pp. 58–77, 2002.

[12] L. M. Butler, "A multidimensional approach for analyzing variants of code writing questions in a cs1 course," Ph.D. dissertation, 2019.

[13] K. Duffield and J. Spencer, "A survey of medical students' views about the purposes and fairness of assessment," *Medical education*, vol. 36, no. 9, pp. 879–886, 2002.

[14] M. A. Flores, A. M. Veiga Simão, A. Barros, and D. Pereira, "Perceptions of effectiveness, fairness and feedback of assessment methods: a study in higher education," *Studies in Higher Education*, vol. 40, no. 9, pp. 1523–1534, 2015.

[15] A. Strauss and J. Corbin, *Basics of qualitative research techniques*. Sage publications Thousand Oaks, CA, 1998.

[16] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft similarity and soft cosine measure: Similarity of features in vector space model," *Computación y Sistemas*, vol. 18, no. 3, pp. 491–504, 2014.

[17] S. Graham, "A review of attribution theory in achievement contexts," *Educational Psychology Review*, vol. 3, no. 1, pp. 5–39, 1991.

[18] D. T. Miller and M. Ross, "Self-serving biases in the attribution of causality: Fact or fiction?" *Psychological bulletin*, vol. 82, no. 2, p. 213, 1975.

[19] J. G. Simon and N. T. Feather, "Causal attributions for success and failure at university examinations." *Journal of Educational Psychology*, vol. 64, no. 1, p. 46, 1973.

[20] N. Hawi, "Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course," *Computers & Education*, vol. 54, no. 4, pp. 1127–1136, 2010.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.