Developing Accessible and Equitable Remote Exams at Scale During the COVID-19 Pandemic

Peyrin Kao
Nicholas Weaver, Ed.
Dan Garcia, Ed.

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-223
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-223.html

August 20, 2022

Acknowledgement

We would like to acknowledge the dedicated teaching staff who helped support students and design remote exams throughout the pandemic, including: Ben Hoberman, Caroline Liu, Ethan Ordentlich, Fuzail Shakir, Jero Wang, Jerry Xu, Justin Yokota, Madison Bohannan, Nicholas Ngai, Ryan Lehmkuhl, Shomil Jain, Vron Vance, Zephyr Barkan, and many others.

Developing Accessible and Equitable Remote Exams at Scale During the COVID-19 Pandemic

by

Peyrin Kao

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Nicholas Weaver, Co-chair
Dan Garcia, Co-chair

Summer 2022

Developing Accessible and Equitable Remote Exams at Scale During the COVID-19 Pandemic

Abstract

Developing Accessible and Equitable Remote Exams at Scale During the COVID-19 Pandemic

by

Peyrin Kao

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Nicholas Weaver, Co-chair

Dan Garcia, Co-chair

For most of 2020 and 2021, schools and universities around the world switched to remote instruction formats as a result of the COVID-19 pandemic. In particular, this meant that in-person assessments had to be adapted into an online format for remote delivery.

This report explores and documents various approaches to large-scale remote exams taken by the Electrical Engineering and Computer Science (EECS) department at UC Berkeley. Using both qualitative feedback from students and instructors and quantitative feedback from surveys and exam data, we analyze the benefits and drawbacks of several exam formats, exam delivery platforms, remote proctoring policies, and remote cheating detection strategies. We recommend a set of best practices and guidelines for large-scale remote exams and hybrid exams (where students can take the exam either in-person or remotely).

Well-designed remote exams have made significant progress in improving accessibility, flexibility, and stress reduction for students. Although classes are starting to transition back to in-person formats, we hope that these developments in remote exams can be retained and incorporated into future exams to create a more equitable learning environment for all.

Contents

Acknowledgments

We would like to acknowledge the faculty members who offered invaluable input as we designed new policies to adapt to online learning, including Dan Garcia, Michael Ball, Nicholas Weaver, and Adam Blank.

We would also like to acknowledge the dedicated teaching staff who helped support students and design remote exams throughout the pandemic, including: Ben Hoberman, Caroline Liu, Ethan Ordentlich, Fuzail Shakir, Jero Wang, Jerry Xu, Justin Yokota, Madison Bohannan, Nicholas Ngai, Ryan Lehmkuhl, Shomil Jain, Vron Vance, Zephyr Barkan, and many others.

Chapter 1

Remote Exam Formats

1.1 UC Berkeley EECS Class Structure

Typical class sizes in the Electrical Engineering and Computer Science (EECS) department at UC Berkeley range from 50-150 students in more specialized upper-division classes, 400-800 students in larger upper-division classes, and 1000-2000 students in lower-division classes. Because of these large class sizes, EECS classes have adopted policies and developed infrastructure specifically designed for scalability [30]. When the outbreak of COVID-19 forced schools to abandon in-person instruction in March 2020, the scalable policy and infrastructure of EECS classes helped facilitate the transition to remote learning.

For example, to allow enrolling more students than the lecture hall capacity, almost all large classes record lectures and post them online for students to watch later. As a result, students who were already accustomed to watching recorded lectures before the pandemic were minimally disturbed by the shift to remote lecture formats during the pandemic. Before the pandemic, some classes had also experimented with remote and recorded discussion sections and labs, which helped to streamline the conversion of all discussion sections to the remote format.

Additionally, the grading infrastructure for most EECS classes has been designed to be scalable for very large class sizes. Most classes use Gradescope to grade assignments, which allows for autograding many submissions and streamlining the manual grading process.

Despite the scalable design of some larger classes, the large class sizes in EECS do present some unique disadvantages when it comes to transitioning to a remote format. Most notably, many strategies for delivering remote exams, such as oral exams, scale poorly to classes with hundreds or thousands of students [39]. Prior to the pandemic, all EECS classes exclusively delivered exams in-person. Even classes that were designed to be online, such as CS W186 (Introduction to Database Systems), only offered in-person exams prior to the pandemic [46].

When the pandemic began, classes had to develop new remote exam formats. Most classes opted to stay close to the existing exam structure, but a few classes changed their exam format or replaced assessments entirely. For this report, we surveyed and interviewed staff from most large undergraduate EECS classes (both lower-division and upper-division) to understand the different remote exam structures used throughout the pandemic. The remote exam formats developed by classes can be separated into a few broad categories, described in the following sections. The structure of in-person exams is also described below as a point of reference.

1.2 In-Person Exam Formats

Overview

Before the pandemic, almost every large EECS class delivered synchronous, in-person paper exams. All students took the exam in large rooms at the same time, and were allowed to work on the entire exam during the allotted time.

If the class size was too large to fit in a single room, the exam would take place simultaneously in different rooms across campus. Some classes sorted students into exam rooms by groups (e.g. last name, student ID number) and allowed students to sit anywhere in their assigned room. Other classes used a seating tool web app developed in the department to assign students randomly to a specific room and seat, as shown in Figure 1.1. If needed, students could request seating accommodations (e.g. left-handed desk) before being assigned to a seat.

Exam Content

Students filled out the exams on paper, but the answer format usually differed by class. For example, theory-focused classes like CS 70 (Discrete Mathematics and Probability Theory) and CS 170 (Algorithms) require students to write mathematical proofs. Introductory classes like CS 61A (Structure and Interpretation of Computer Programs, i.e. CS 1) and CS 61B (Data Structures, i.e. CS 2) have coding questions where students must write a code snippet or fill in the blanks in some code scaffolding. Software classes like CS 161 (Computer Security) and CS W186 (Database Systems) have multiple-choice and short-answer exams. After the exam, the paper exams are scanned and uploaded to Gradescope, where they are graded (automatically or manually, depending on the question) and grades are released to students.

A few classes, such as CS 10 (Beauty and Joy of Computing), also assigned computer-based exams, where students use part of a 2-hour lab section to write coding questions in Snap*!* and Python on a computer.

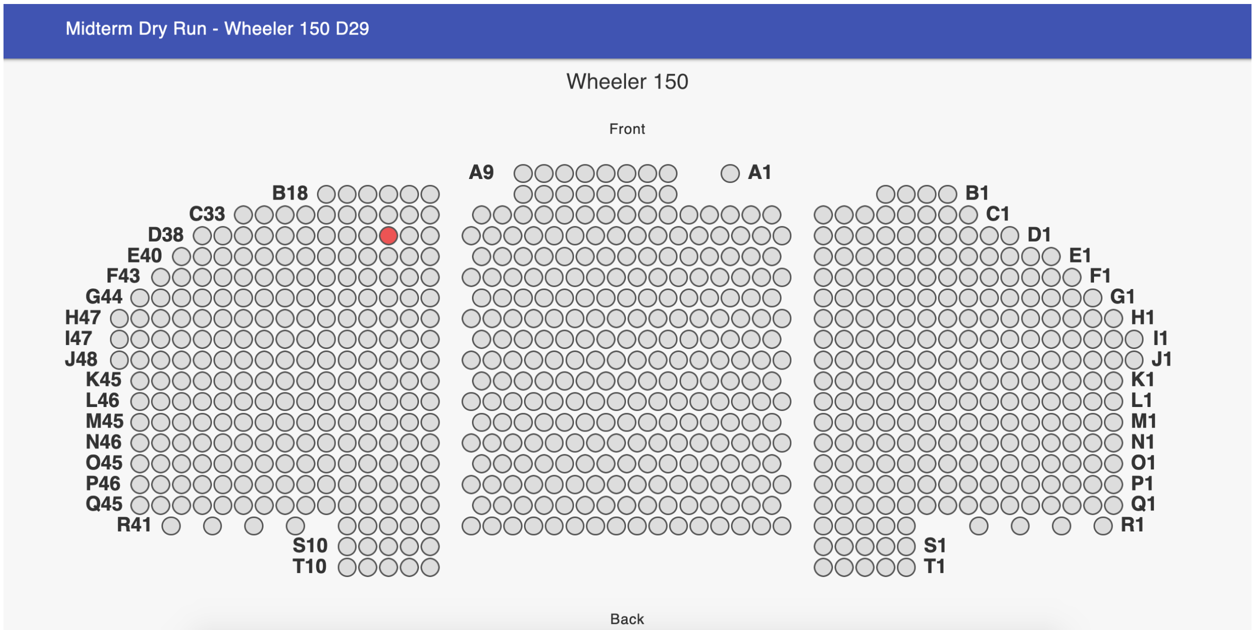


Figure 1.1: Student view of the seating assignment tool showing the student's assigned seat during the exam.

Allowed Resources

Exams could be either closed-book or open-book, but electronic devices such as smartphones and laptops are not allowed. One common policy was allowing a limited number of handwritten note sheets (commonly called "cheat sheets") during the exam.

Clarifications

Most exams allowed students to ask course staff clarification questions if a question on the exam was unclear or ambiguous. If staff decide to answer the question, the clarification is displayed to all students. A live-updated clarification document (e.g. Google Docs) was usually displayed on a projector at the front of the room. For fairness, questions are not supposed to be answered individually; a question is either answered with a clarification to all students, or not answered at all. Sometimes, staff members will directly answer minor questions like typos or students forgetting to read the question, but this is informal and not usually documented in course policy.

For large classes, students may be distributed across several different classrooms or lecture halls. A simple setup for distributed clarifications is to have staff in each room process clarifications independently and communicate online (e.g. through Slack) when a clarification requires more discussion and possibly clarification to the entire class.

Some classes did not allow any clarifications, instead opting to grade questions differently after the exam if an error was discovered in the question.

Timing Accommodations

Classes have the freedom to set their own midterm times. Because class sizes often exceed lecture hall capacity, midterms are often held in the evenings (e.g. 7:00 PM–9:00 PM) when additional lecture halls can be used. Final exam times are set by the university and are usually based on the lecture time to minimize time conflicts for students. However, because EECS classes record lectures and usually let students enroll in classes with conflicting lecture times, students may encounter time conflicts with final exams. For students who cannot attend the exam at the designated time, there are both formal accommodation policies from the university and informal accommodation policies used by most classes in the department.

The university has an official policy for student athletes who cannot take an exam because they have a time conflict with a university-sanctioned event. If an alternate exam time cannot be arranged for the student, the university allows an approved member of athletics staff to serve as exam proctor while the student takes the exam away from campus [47].

The university also has an official policy for students to reschedule their exam if the scheduled exam time violates the student's religious beliefs. The official policy does have a statement for potential exceptions to rescheduling if "administering the examination at an alternate time would impose an undue hardship which could not reasonably have been avoided," but also stresses that "reasonable common sense, judgment and the pursuit of mutual goodwill should result in the positive resolution of scheduling conflicts." [38] These situations are usually resolved on a case-by-case basis, though if a religious observance affects a larger group of students, an announcement may be made in advance for students to reach out privately.

Other reasons for taking an alternate-time exam are usually handled differently by different classes. The most common of these reasons is a time conflict with another exam; the instructor of the class usually decides whether to allow exam time conflicts at the beginning of the semester. Many classes handle situations such as family emergencies, travel, and mental health conditions on a case-by-case basis.

If a class does choose to offer accommodations, they usually come in the form of an alternate exam time that happens directly after the scheduled exam. The same exam, possibly with some minor versioning changes, is usually given in both sittings. For example, if the main exam is 5:00 PM to 7:00 PM, the alternate exam might begin at 6:00 PM or 7:00 PM. The lack of gap between exams is designed to ensure that students from the main exam do not have an opportunity to discuss the exam with students taking the alternate exam. Because exams often happen in the evening to avoid conflicts with other classes, the alternate exams sometimes run as late as midnight. Sometimes alternate exams begin before the main exam (e.g. for a 5-7 exam, the alternate is 3–5 instead of 7–9) in order to avoid finishing exams too late at night, which inconveniences students and may cause safety concerns with students getting home late at night.

Most classes have informal policies for students who fall ill or otherwise have some emergency during a scheduled exam time. These are usually handled individually on a case-by-case basis.

If the student cannot take any reasonable alternate-time exam, some classes may offer the option to skip the exam. For midterm skips, the final exam can usually be used to replace the student's midterm score (scaled appropriately to reflect exam distributions, e.g. by z-score), a term called "clobbering" within the EECS department. For example, a student who misses midterm 2 may have their score for that exam replaced by their final exam score. The score is usually adjusted (e.g. by z-score) to account for different exam difficulty. The missed exam score is usually replaced by a later exam score, so it is difficult to clobber the final exam. Some classes might allow the student to take an Incomplete grade, miss the final, and take the final exam in a later semester.

DSP Accommodations

The Disabled Students' Program (DSP) at Berkeley works with students to arrange accommodations in classes where their disability might put them at an unfair disadvantage. The DSP office offers several accommodations for exams. The most common accommodation is extra time, where students are given 150% or 200% of the normal exam time to complete the exam. For example, a student with a 150% accommodation is given 4.5 hours to complete a 3-hour exam.

Another common DSP accommodation is "reduced distraction," which means the student should be allowed to take the exam in a quiet environment. Large exam rooms with hundreds of students rarely meet this criteria, so students with the reduced distraction accommodation often take the exam in a smaller, quieter room at the same time.

Officially, a reduced distraction room meets the DSP requirements if it has minimal movement and is not filled above 1/3 capacity, which is only slightly less dense than the departmental norm during exams. But most classes when providing a low distraction environment also seek to limit the number of students per room, a criteria not present in the official definition.

Less common exam accommodations include "daytime exams," which means the student must finish exams before 5:00 PM, "rest breaks," which means the student is allowed periodic rest breaks during the exam, and "one exam per day," which means the student cannot take multiple exams on the same day (this is especially relevant for final exams).

Because of the extra time and reduced distraction accommodations, DSP students often take the exam in a separate, smaller room, where the exam is proctored beyond its designated end time. (For example, for a 5:00 PM to 8:00 PM exam, the DSP room would not be finished until 11:00 PM.) If the exam end time is too late, sometimes the DSP exam will start earlier (in the example, the DSP exam might be from 2:00 PM to 8:00 PM). This room is sometimes also used for students taking an alternate time exam.

There are also DSP accommodations that relate to equipment instead of exam scheduling, such as large-print exams (larger font size), use of earplugs, use of an adjustable table, and use of a computer to input exam answers (possibly with voice output and/or voice input) [1].

1.3 Synchronous Online Exams

Overview

The most commonly-used remote exam format is the synchronous online exam, which most closely follows the existing in-person exam format. Synchronous online exams differ from the asynchronous formats below in that students are expected to start and finish the exam at the same time.

Exam Content

Synchronous online exam questions are most similar to in-person exam questions (compared to the other formats listed below), in part because synchronous online exams are designed to be taken in the same amount of time as an in-person exam.

However, since students can potentially use the Internet during the exam (see the "Allowed Resources" section below), the questions may need to be written in a way that precludes searching the Internet directly for the answer. For example, "What is the runtime of mergesort?" might be an appropriate question on an in-person exam, but for an online exam, students can simply look up the right answer [5]. Writing these "Google-resistant" questions may result in these exams being more difficult than their in-person counterparts, even though the exam content is similar.

Additionally, synchronous online exams may be longer than their in-person counterparts to counteract cheating. If an online exam is too short, students might finish early and use the remaining time to share answers with others [13]. Some classes deliberately lengthened their exams when switching to a synchronous online format to stop students from finishing early. However, overly long exams potentially have equity issues with students who are slower test-takers, since exams are usually not designed to test speed. Long exams may also incentivize test-taking strategies such as optimizing by solving easier questions that are worth more points first. Finally, the lower grade distributions for an exam that is too long for most students to finish may also necessitate an adjustment in grading policy (e.g. using a curved grading formula).

Allowed Resources

As with in-person exams, synchronous online exams could be either open-book or closed-book, but additionally, online exams have the option of being open-Internet. Students could be allowed to access online class resources such as lecture slides or past exams, or search the Internet for other resources.

Many open-Internet exams came with the restriction that students were not allowed to directly look up answers. Students were sometimes also explicitly prevented from reposting exam questions on Q&A platforms such as Chegg or Stack Overflow (to try and solicit answers during the exam) [29]. Most open-Internet exams came with the restriction that students could not use the Internet to collude with other students during the exam, although a few classes allowed collaborating with other students through moderated platforms such as Piazza.

There have been studies comparing closed-book, cheat-sheet, and open-book exams to determine student attitudes toward each type of exam and how each type of exam facilitates student retention of material [20]. With the advent of open-Internet exams, these studies could be extended to see how exams are affected when students are allowed to freely consult any resource on the Internet. One belief is that open-Internet exams are a better reflection of real-world programming skills, because programming in the real world often includes looking up documentation, troubleshooting, and searching for existing solutions to common problems. Open-Internet exams may also lower test anxiety for students, because they do not have to stress about accidentally opening a disallowed resource during the exam.

Clarifications

Initially, while classes were still trying to adapt to the remote exam format, some exams did not allow clarification requests, since the infrastructure was not in place yet. Later, as more robust infrastructure for remote exams was developed, more classes allowed clarification requests and answers.

Online exams have significantly reduced the barrier to asking clarification questions. Asking a question during an in-person exam usually requires a student to raise their hand and find a TA, which can be difficult in a crowded exam room. However, during online exams, asking for clarification can be as simple as typing a question into a form and submitting it. This has resulted in a significant increase in the number of clarification requests in online exams, as shown later in this report.

The remote exam infrastructure makes it possible to answer clarification questions, either individually or as announcements to the entire class. While most classes opted to only answer questions via announcement or not at all, a few classes did answer clarification questions individually. This shift in policy could be because of the larger number of clarification questions received during the exam, which would make the list of public clarifications excessively long.

Timing Accommodations

Because of the pandemic, there were more circumstances where students were unable to take the exam at the scheduled time. Unlike in-person exams, where the vast majority of students are in the same time zone, many students went home to different time zones during the pandemic. However, synchronous online exams require every student to start the exam at the same time, which means that students in different time zones may have to start the exam at inconvenient times.

Additionally, the number of medical-related timing accommodations increased because of COVID-19. Students requested alternate times in order to get tested, get vaccinated, help a family member with COVID-19, or recover from COVID-19 themselves.

One alternate exam directly after the scheduled exam was often not enough to accommodate all time zones. Some classes addressed this by providing multiple alternate times. Others announced the alternate time before the semester and advised students to take a different class if their time zone was not compatible with the scheduled exam times. A few classes offered no alternate exams at all, requiring students to take the class only if they could take the exam at the designated time.

From an exam security perspective, alternate exams can lead to exam leakage. Scheduling in-person alternate exams directly after the main exam is usually sufficient to prevent exam leaks between exam times, especially since students are not allowed to keep a copy of the exam to share. However, in synchronous online exams, it is easier for students to save and distribute a copy of the exam to students taking the alternate exam. Even if the main exam time overlaps with the alternate exam time, as soon as one student has received a copy of the exam, it is possible to spread the exam to other students.

To avoid exam leakage, most classes only scheduled alternate exams to start after the main exam. This still runs the risk of students from the main exam time sharing the exam with students taking the alternate exam. However, this is still preferable to scheduling alternate times before the main exam, where a few students taking the alternate exam could potentially leak the exam to the majority of the class taking the main exam before the main exam starts.

Because of the security implications of the exam leaking before the alternate times, many classes only allowed students to take alternate exams if they had a valid reason, such as a time zone conflict or a medical emergency. However, some classes did offer alternate exams to all students, with no justification required. Generally only a small proportion of students signed up for the alternate exams.

Online exams are no longer dependent on room bookings, so classes had more flexibility in scheduling exam times. However, in order to accommodate as many students as possible, classes had to schedule midterm exams with other time zones in mind. Some classes sent out surveys at the beginning of the semester to assess what time zones students were in and what exam times would work best for everyone. Because our introductory surveys (such as the one in Figure 1.2) showed that most students were in the United States and east Asia (India, China, Taiwan, Japan, Korea), an especially popular exam time was the early evening in the western US, which corresponded to the morning in east Asia. This time may have also been preserved from in-person exams, where midterms were often scheduled in the early evening to avoid conflicts with other classes and ensure adequate room bookings. Other classes used their existing lecture time for exams.

Classes had less flexibility to schedule final exams, since the university still assigned final exam times during the pandemic. Final exams are scheduled between 8:00 AM and 10:00

PM in Pacific time, which corresponds to approximately 11:00 PM to 1:00 PM in east Asia. This caused issues for students in east Asia who had to request alternate times or take exams in the middle of the night.

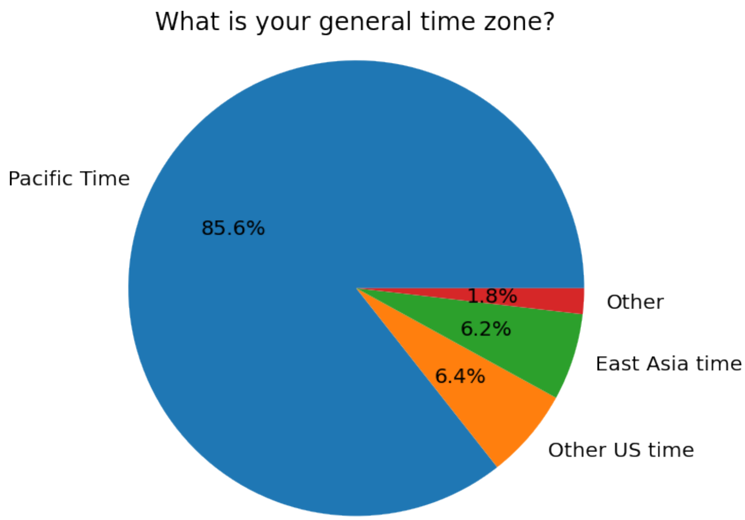


Figure 1.2: Percentage of CS 161 students (upper-division computer security class of 450 students) in the same time zone as the university during Fall 2020, a fully-remote semester.

DSP Accommodations

Classes also had to schedule exam times for DSP students with extra time accommodations. The combination of scheduling more alternate exams in addition to DSP extra time exams resulted in more complex exam schedules compared to in-person exams.

Scheduling longer DSP exams required balancing exam security and avoiding inconvenient times. In some cases, since exams were late at night, DSP exams started earlier in order to avoid finishing too late, despite the risk of exam leakage.

1.4 Per-Question Synchronous Online Exams

Most synchronous exams were timed like in-person exams: students were given the entire exam and had the entire allotted time to finish the exam in any order. Students could freely skip between questions or go back to previously completed questions, as in an in-person exam. One variation on synchronous online exams was timing exams by question: students were given one question and a fraction of the exam time to complete it. During this time, students could only work on that one question and could not work ahead or go back to a previous question. Once the time expired, they were given the next question and another block of time to work only on that question.

From an exam security standpoint, this structure makes it more difficult for students to collude on a question [13]. This idea can be strengthened by giving students the questions in random order (discussed in more detail later), so that in any given block, two students are probably working on different questions and cannot directly share answers.

This format requires some additional considerations that are not needed on a standard synchronous online exam. When writing the exam, each question must be assigned its own time limit. This can be problematic because different students may need different amounts of time to solve each question, and by enforcing a per-question time limit, students lose the flexibility to borrow time from one question that they finish early and spend that time on another question that they find harder. If questions are delivered to students in random order, the questions may additionally need to be calibrated to take the same amount of time to complete. Also, additional infrastructure must be built for delivering each question to students at the designated time and preventing students from going back to work on past questions.

1.5 Asynchronous Online Exams

Overview

In asynchronous online exams, students were given a long time window (often 24–72 hours) to complete the exam. The exam is designed to take significantly less time to finish (usually 2–3 hours, like a normal in-person exam), but students are given the flexibility to decide when they want to work on the exam and how much time they want to spend on the exam. Some students chose to finish the exam in one sitting, while others used several sittings across the exam period to finish the exam.

Exam Content

The more flexible time limit for asynchronous exams was reflected in the exam content. Some exams were written with a similar structure as in-person exams, while other exams involved more practical coding tasks that require additional allowed resources (see below).

Allowed Resources

Because students had a large time window to complete exams, asynchronous exams were usually open-book, so that students could refer to resources during the exam window. It would be rather unrealistic (and possibly encouraging cheating) to require that students avoid looking at any class resources or the Internet for several days while the exam was available.

When the exam content was more hands-on, additional resources (besides content resources like class notes or lecture slides) were often recommended or even required to finish the exam. For example, if the student was required to write code to be executed, an IDE and a compiler would be allowed.

As in synchronous exams, directly looking up answers, reposting questions online, and collaboration with other students was usually explicitly forbidden. However, because of the long time window, it was difficult to enforce this.

Clarifications

Unlike synchronous exams, where clarifications are all submitted and processed during a set exam time, asynchronous exams that offered clarifications would have to process clarifications throughout the entire 24-72 hour exam window. Although this meant that clarifications came in at a slower rate, it also meant that either staff would need to process clarifications constantly across the exam window, or some overnight clarifications would go unanswered for a few hours. Additionally, students who worked on the exam earlier in the time window might miss clarifications that were announced later in the exam window.

Timing Accommodations

The long window for asynchronous exams inherently solves most of the timing conflicts from synchronous or in-person exams. Students with a time conflict could choose to take the exam in any part of the 24-72 exam window that worked best for them.

However, there is a trade-off between flexible timing and exam security. Although exam leaking is not a concern, since everyone can access the exam in the time window, there is the risk of one student completing the exam, and then passing the answers on for a student taking the exam later.

DSP Accommodations

Because the time window for asynchronous exams was much longer than the intended time needed to finish the exam, some classes did not offer DSP accommodations, with the reasoning that a student could take 150% or 200% of the intended 2-hour time to finish the exam and still finish within the time window. Another approach is to reason that if students are given 24 hours to finish the exam, then a DSP student should be given a 36-hour or 48-hour window instead.

1.6 Limited-Time Asynchronous Online Exams

A slight variation to this format limited students to a fixed time limit, similar to synchronous online exams. The student could start the exam anytime within the time window, but once they started the exam, they had to finish and submit the exam within the allotted time. For example, on a 2-hour midterm, the student could start at 1:00 PM, but would be required to complete the exam by 3:00 PM.

1.7 Alternate Assessment Methods

Overview

Some classes chose to replace exams with a different assessment, such as a project or report. Students were given a long window (days or weeks) to complete a longer assignment to demonstrate their understanding of the material.

There are also alternate exam formats such as an oral exam [39], but none of the large classes in EECS tried these approaches, possibly because of difficulty with scaling these approaches to large classes. However, oral exams were used as a supplement to cheating detection, as discussed in the cheating detection section.

Assessment Content

Most, if not all EECS classes gave exams before the pandemic, so the classes that chose to use alternate assessments had to develop new assignments from scratch. These assignments were often open-ended and asked students to explore new topics using concepts from the class, rather than asking all students to converge on one correct solution. This type of assessment was mostly adopted by classes outside of EECS, but a few EECS classes were able to design alternate assessments.

Allowed Resources

Since the time window was so long, these assignments were almost always open-book and open-Internet. Some classes may have allowed students to work in small groups, or invited students to talk with fellow students or TAs for guidance. Since the assignments were more open-ended and did not have one correct solution, most of the cheating and exam leaking concerns from remote exams were less applicable.

Accommodations

Alternate assessments were structured more like projects than assessments, so most of the exam-related accommodations did not apply here. Instead, classes had to handle some project-related accommodations. For example, DSP students are sometimes granted additional days to work on projects, so their due date for the assessment was extended by a few days.

Chapter 2

Remote Exam Platforms

To support remote exams, EECS classes adapted existing infrastructure to support online exam delivery. Later during the pandemic, several new platforms for online exam delivery were developed as well. A summary of the benefits and drawbacks of each remote exam platform can be found in Figure 2.1.

2.1 Gradescope

Most EECS classes were already using Gradescope for assignments and exams before the pandemic, so Gradescope was the most familiar assessment platform for staff and students at the start of the pandemic.

Gradescope supports two main ways for students to submit exams. The online assignment feature lets students enter multiple-choice or free-response questions into radio boxes, check boxes, and text boxes. The multiple-choice questions can be autograded: Gradescope supports all-or-nothing grading on multiple-choice questions, and partial credit grading on multiple-choice questions can be automated with some Javascript. This method was mostly used by exams with multiple-choice, short-answer, or coding questions. Initially, students had to manually click "Save" after each answer, but later in the pandemic, the "timed assignment" feature automatically saved answers.

The exam upload feature lets students upload a PDF or images of their exam submission. Students would write their exam answers on a printout of the exam (or a blank sheet of paper), scan or take photos of their answers after the exam, and then upload them to Gradescope. This method was mostly used by exams where answers required mathematical notation or diagrams, which cannot be directly typed into Gradescope. Students were usually given 5-15 minutes after the exam was over so they could upload scans of their exam. However, there were still technical issues involved in the submission process.

The major drawback of Gradescope is the lack of support for automatically generating different versions of exams (as of 2022). Generating different versions of exams with randomization and versioning improves exam security, as discussed later in the cheating detection section.

Some classes found temporary workarounds to create exams with different versions. For example, CS 161 (Computer Security) created a blank answer sheet on Gradescope with only letters as answer choices. Each student received a customized exam PDF with their version and matched the labeled answers on their PDF with the labels on the Gradescope answer sheet. External Python scripts were used to create randomized PDFs and derandomize the submissions to the Gradescope answer sheet. This allowed students to receive different versions of the exam while using an existing, familiar platform for submitting answers.

Gradescope has scalable computing power, but can be overwhelmed if too many people log on at the same time. This affected at least one large class at Berkeley trying to use Gradescope for online exams, and seems to have affected classes at other universities as well [17].

2.2 Canvas

UC Berkeley provides the open-source Canvas learning management system to all classes as bCourses. Canvas is similar to Gradescope in that it allows students to either enter answers directly into their browser with the quiz feature. Canvas supports automatically randomizing questions in a quiz for better cheating prevention, but the randomization may not be fully customizable. Canvas supports providing a question bank from which students receive a random subset of questions, but does not support dynamic question generation. Canvas also supports uploading answers as a file. However, most classes opted for Gradescope if their exam required a file upload at the end of the exam, since students and staff are more familiar with using Gradescope for online manual exam grading.

The Canvas backend used by UC Berkeley is scalable to some extent, but was also overwhelmed by at least one large class using Canvas for online exams. Another minor downside to Canvas is that most EECS classes have traditionally used Gradescope for assignments. This meant that students might not be familiar with Canvas and that exam scores from Canvas had to be transferred to Gradescope or some other central source of grades.

2.3 CS 61A Exam Tool

In mid-2020, the staff from CS 61A developed a new browser-based exam delivery platform customized for specific needs in EECS. This platform, colloquially known as “Exam Tool,” was adapted by most of the EECS department for online browser-based exams. Staff from other classes also helped contribute development and features to Exam Tool throughout the pandemic.

In response to the scalability issues from Gradescope and Canvas, Exam Tool used a backend that was able to scale quickly and effectively with thousands of students using the platform.

Like Gradescope and Canvas, Exam Tool supports radio boxes, check boxes, and text boxes for students to enter answers to multiple-choice, short-answer, and coding questions. Course staff use a custom syntax based on Markdown to edit exams and input questions with answer choices, short-answer boxes, or coding boxes.

Exam Tool supports customized randomization so that each student receives a different version of the exam. Using some custom syntax, staff can specify when they want questions, subparts, or answer choices to be randomized. Staff can also specify different versions of questions to supply to students, or provide a question bank from which students receive a random subset of questions.

Exam Tool supports watermarking and keylogging to help detect cheaters, as discussed later in the cheating detection section.

Another important feature of Exam Tool is producing encrypted PDFs. Exam Tool can create two separate versions of student's randomized exam, a web-based exam and a PDF exam. The PDF form would be encrypted and emailed to the students in advance, with the decryption key provided at the start of the exam, allowing a student with a problematic Internet connection to still take the exam by printing the PDF and photographing the results.

Exam Tool does not natively support exam grading. Instead, student submissions are derandomized into a consistent form, exported as PDFs, and uploaded to Gradescope for grading. This allows exams to be graded using Gradescope's streamlined grading and autograding features, and it allows exam scores to be stored in the same place as other assignment scores for grade processing.

2.4 PrairieLearn

PrairieLearn is an online question generation and delivery platform designed for proficiency-based learning. It can be used to build question generators, which automatically generate random variants of questions for students to try until they have sufficiently practiced and learned the material [48].

Question generators can be useful for building and delivering scalable, randomized exams [50]. Once the question generator is written, PrairieLearn can automatically deliver a different random variant of the question to every student. PrairieLearn also supports randomizing the order of answer choices, question subparts, or entire questions. In an exam setting, question generators must be carefully written so that different variants of a question are of equal difficulty for students. This is discussed in more detail in the exam randomization section later in this report.

PrairieLearn questions can be written for automatic grading, and the platform natively supports real-time grading of student work. For remote exams that try to emulate the experience of in-person exams, real-time grading can be disabled so that students do not see whether their answer is correct or not as they are taking the exam. The question can still be graded automatically, but the student will not see the results until after the exam is over.

	Gradescope	Canvas	Exam Tool	PrairieLearn
Input answers in browser	×	×	×	×
Upload handwritten answers	×			
Randomize answer choice order		×	×	×
Randomize order of all questions		×	×	×
Randomize order of selected questions			×	×
Versioning with keywords			×	×
Supports grading	×	×		×
Did not crash for large classes			×	×

Figure 2.1: Comparison of features on various remote exam platforms.

The main drawback of PrairieLearn for use in the EECS department was that students and staff had not used the platform before, so there was a steeper learning curve for PrairieLearn compared to a more familiar platform like Gradescope. In particular, writing a question generator requires specifying variations on the question that can be randomly generated, and many instructors did not have any prior experience with question generators.

One long-term benefit of PrairieLearn and question generators in general is the ability to create a new variant of an exam on-demand, which can be helpful for scheduling alternate-time exams and increasing flexibility for students. Question generators also allow exam questions to be reused across semesters, freeing up staff resources that are typically used to write a new exam every semester. Although only a few classes tried writing exams with question generators during the pandemic, further work on question generators can facilitate more widespread adoption of these tools.

Chapter 3

Remote Exam Proctoring

3.1 Timeline of Proctoring Guidelines

Before discussing the development of various policies within EECS, it is useful to first understand the timeline of guidelines and restrictions on remote proctoring released by the university during the pandemic. These restrictions affected the development of various policies, and some policies were developed but never used as a result of these guidelines. Additionally, some of these guidelines were released in response to feedback to the proctoring policies developed by various classes.

When classes were cancelled for the rest of the spring semester on March 10, 2020 [10], no proctoring guidelines from the university had been finalized yet, so classes were allowed to develop proctoring policies without any restrictions. On March 27, 2020, the university banned all forms of remote proctoring for the rest of the spring 2020 semester, citing privacy concerns. This forced several classes to abandon their proctoring policies and left all exams for the rest of the spring 2020 semester unproctored [3].

On June 12, 2020, the university established a pilot program in which classes were allowed to use the Zoom video conferencing tool for remote exam proctoring for their final exams in the summer 2020 semester [2]. (Note that enrolling in summer semesters at Berkeley is optional.) Classes had to follow a set of guidelines to ensure that students were not disadvantaged as a result of the proctoring policy. Additionally, the policy allowed for classes to use Zoom to proctor final exams for the fall semester as well, but only if the department first sent a request to the university, specifically the Vice Chancellor for Undergraduate Education (VCUE). The policy announcement did not specify whether this policy extended to midterm exams, which caused some confusion.

Shortly after this announcement, instructors from the EECS department contacted the department chairs, who in turn reached out to the university regarding remote proctoring approval for both summer and fall semesters. On June 23, 2020, the EECS department received approval from the university, specifically the office of the Vice Chancellor for Undergraduate Education (VCUE) and the Executive Vice Chancellor and Provost (EVCP), to remotely proctor all exams (midterms and finals) in the summer 2020 and fall 2020 semesters. As part of this agreement, EECS classes were asked to report any cheating cases to the Office of Student Conduct (as required by campus policy) so that students would receive due process on any cheating allegations, and to provide regular updates to the EVCP office, the VCUE office, and the Academic Senate on exam cheating. Additionally, classes were advised to move away from curved grading and toward "equitable grading structures." This approval occurred through personal email communication and was distributed privately to the instructors of summer classes. This led to the development of many different proctoring policies for the summer and fall semesters.

On December 7, 2020, the pilot program was extended so that EECS classes could continue remote proctoring through the spring 2021 and summer 2021 semesters [28]. Many of the policies developed in fall 2020 were continued into the spring 2021 semester.

On July 12, 2021, the university authorized Zoom proctoring for all classes in fall 2021 as part of a broader plan to accommodate hybrid instruction (classes taught both in-person and remotely) [36]. This allowed existing remote proctoring policies to continue being used in hybrid exam models, which are discussed later in this report.

3.2 Motivation for Remote Proctoring

The most prominent rationale behind implementing remote proctoring is cheating detection and prevention. Past studies have shown that cheating rates are higher when remote exams are unproctored [21].

There are several factors behind these higher cheating rates. Students have reported that they are more likely to cheat and that they feel cheating is more acceptable when exams are online, and especially when exams are online and unproctored [15, 40]. Additionally, students who see their peers cheating may feel that it is rational for them to cheat as well, creating a feedback loop that increases cheating rates [34]. Studies have shown that when given multiple unproctored exams, the cheating rate increases as more students are tempted to cheat and students learn effective cheating strategies [8].

Almost all forms of remote proctoring provide strong prevention benefits over an entirely unproctored exam. Students who think that they are being watched, or that evidence of their cheating will be logged, are less likely to cheat, even if they are not actually being watched or if the evidence is not actually reviewed [15]. Remote proctoring also gives other students better assurances about exam security; knowing that everyone is being proctored makes it less likely that others are cheating, which in turn reduces incentives from peers to cheat [34]. Some forms of proctoring can also provide detection if they are designed to catch students trying common forms of cheating. Cheating detection is discussed in more detail later in this report.

A secondary reason for remote proctoring is to give instructors a continuous communication channel with students during the entire exam. Some instructors outside of EECS did not require students to send any audio or video feeds to the instructor, and instead only used Zoom as a tool to communicate announcements to students and allow students to ask questions during the exam.

3.3 In-Person Exam Proctoring

Before discussing remote proctoring, as a point of comparison, it is useful to consider the existing in-person exam proctoring strategies for large EECS classes, as well as possible methods of cheating under this proctoring.

Each exam room has a number of proctors (usually TAs and instructors) and students. The ratio of proctors to students varies depending on class size and the size of the exam room, but is usually on the order of magnitude of around 10 to 100 students per proctor. Proctors are either managing logistics in the front of the exam room or walking around to answer clarification questions. Because exams are usually held in large lecture halls, this also means that many students are not being actively monitored by a proctor at any given time.

Students usually either sit in adjacent seats or alternating seats. There is usually only a single version of the exam written for an in-person exam, so reading a neighbor's exam would be a viable method of cheating. Some in-person exams did have multiple versions, but for ease of grading, all the students in one room sometimes received the same version of the exam, so it would still be possible to cheat off a neighbor. The different versions would be used for later alternate-time exams, as described below.

Occasionally a class would experiment with a versioned exam, where there were 2 to 4 distinct versions. The variation between the exams was limited, usually consisting of simply keywords on one or two questions while maintaining the order of questions and exact format for scanning, in an attempt to detect a student who copied a keyword from their neighbor.

One effect of proctoring exams for large classes is that cheat sheets are often not rigorously checked for every student because it would take too long or inconvenience students taking the exam. This means that students may be able to use extra cheat sheets or disallowed cheat sheets (e.g. typed sheets when only handwritten sheets are allowed) during the exam. However, the value of extra cheat sheets may not be too significant [44], which would make this form of cheating less effective compared to the other strategies described in this section.

If a student needed to use the bathroom, the typical policy is to ask the student to bring their exam and their smartphone to the front of the exam room, where a staff member would keep them while the student left for the bathroom. The time when the student leaves and returns may be logged to check if someone is using the bathroom for unusually long. In some exams, a staff member would also walk the student to the nearest bathroom, or a staff member would be stationed at the nearest bathroom to monitor students. However, for obvious privacy reasons, staff did not monitor students after they went inside the bathroom. To prevent collusion, staff usually only allowed one or two people to use the bathroom at a time. (In the case of two people, staff usually allowed one person to use the men's bathroom and one person to use the women's bathroom.) Although this strictest method precludes most forms of cheating, there is still the possibility of cheating in the bathroom with a second device (e.g. an extra cell phone or a tablet), or leaving answers for a later student in the bathroom. Less strict methods, which are often necessary when there are not enough proctors to supervise every student using the bathroom, might make it possible for students to collude by going to the bathroom at the same time.

After the exam ends, students would be expected to stop writing and pass the exams to the nearest aisle for collection. Sometimes staff would ensure students stop writing by requiring everyone raise their exams in the air, but since exam collection takes a few minutes, and students usually start packing their materials, it is possible to collude during these few minutes.

Finally, there is the possibility of sharing answers between the main exam and the later alternate exam, especially if the exam content is identical. Even if the alternate exam is scheduled immediately after the main exam, a student from the main exam could leave the exam early, or a student from the alternate exam could arrive late, providing a few minutes where they can collude. Sometimes this would be prevented by giving an alternate version of the exam for students taking the exam at different times, though in most cases, the alternate version had the same content as the standard exam, with superficial changes such as changing some numbers. (Changing numbers in different exam versions is described more in the cheating detection section.) Another partial defense against this collusion is preventing students from leaving the exam room when there are less than 10 or 15 minutes left in the exam, though this does not stop students from leaving earlier in the exam. Alternate exam times could also be set to overlap the main exam time (e.g. 5:00–7:00 PM and 6:00–8:00 PM), which would make collusion between exam end and start times difficult, though this may not always be possible depending on the alternate exam times needed.

One possible way to leverage this vulnerability in alternate exams is a student falsely claiming that they need an alternate exam, then colluding with a student taking an exam at the standard time. Some classes may avoid these cheating cases altogether by not providing any alternate-time exams.

In summary, there appear to be four main potential cheating strategies in the in-person proctoring protocol: cheating when a proctor is not looking, cheating in the bathroom, cheating while exams are being collected, and cheating between the main and alternate exams. Previous studies identify similar common cheating strategies [32], though the four strategies listed are more specific to the way large-scale exams are proctored in EECS classes. Remote proctoring strategies can be compared to this baseline.

3.4 Third-Party Proctoring Software

Third-party proctoring software for remote proctoring has been used both before and during the pandemic. UC Berkeley initially did not allow use of this software because of licensing and potential privacy and accessibility concerns. Sometime in 2020, the university approved the use of Honorlock proctoring software for classes in the Haas School of Business, but this did not extend to other departments, including EECS.

A detailed study of this software is beyond the scope of this report, but it is worth noting that third-party proctoring software is usually an external program on the user's computer that requires privileges in order to lock the browser on screen (preventing the user from accessing other websites or applications). This software can also request access to the student's microphone and camera feeds for video proctoring, and the audio and video feeds may be sent to the third-party company. Online proctoring services such as Proctorio, ProctorU, and Honorlock have been criticized during the pandemic for creating stressful exam environments and violating students' privacy [22]. Some of these services use AI to algorithmically detect cheating, which raises additional concerns about biased detection [45]. There have also been reports of students easily circumventing the strict proctoring environments to cheat anyway [19]. These concerns about privacy and biased detection may have contributed to the university's decision to disallow third-party proctoring software.

3.5 Zoom Proctoring Policies

Because of university policy, Zoom is the only video proctoring tool used by the EECS department. Most classes that use Zoom proctoring wrote their own set of policies for how students should use set up Zoom to proctor themselves during the exam.

Devices

Student surveys found that most EECS students had a computer (desktop or laptop) and a smartphone. Some students may have additional devices, such as a tablet, a second computer, or a second phone, but policies tended to assume that students had one computer and one smartphone. There were policies where students were required to use Zoom on only their computer, only their phone, or on both devices.

In most cases, students were not allowed to use their phones during the exam. Requiring that students proctor through their phone has the additional benefit of preventing them from using their phone for other purposes. Alternatively, if students are asked to only proctor through their computer, they might be asked to place their phone face-down somewhere where it is visible.

Zoom Setup

Two general categories of proctoring logistics were developed across the EECS classes during the pandemic. Most classes have opted for a "one-room-per-student" approach, where each student opens a Zoom room of their own and takes the exam.

In contrast, a few classes used a "one-room-per-proctor" approach, where each proctor opens a Zoom room, and many students join each proctoring room. When proctoring a "one-room-per-proctor" exam, proctors watch the live video feeds using Zoom's gallery view, which allows watching multiple video feeds simultaneously. Proctors often have no other responsibilities during the exam except watching video feeds, which helps to ensure that the video feeds are actually watched. (Logistics and clarifications can be handled by a separate team of TAs who don't participate in proctoring.) Proctors can write down any suspicious behavior they notice, such as video feeds disconnecting, students leaving the frame, or screens showing disallowed applications. After the exam, these cases can be cross-referenced with the student's exam for additional evidence of cheating.

In the "one-room-per-student" approach, generally fewer proctoring video feeds are ultimately watched by proctors, because each proctor can watch at most one or two students at a time during the exam. If the video feeds are recorded (see the next section for more details), after the exam, there is usually not enough time for TAs to watch all the recordings. Students may also eventually become aware of this fact, which may decrease the perceived effectiveness of submitting recordings. On the other hand, the "one-room-per-proctor" approach ensures that more proctoring feeds are watched during the exam, but earlier in the pandemic, Zoom did not have a feature to record every video feed in a meeting, so this approach did not have the ability to view recordings afterwards.

The most commonly brought-up security risk of the "one-room-per-proctor" approach is students trying to see answers on other students' video feeds if the video feeds contain exam information. (More details about what the video feeds contain are described below.) This is comparable to students peeking at their neighbor's exam in an in-person exam. However, it turns out that peeking at a neighbor's exam is much harder over Zoom than in-person. Experiments with TAs, as shown in Figure 3.1, showed that Zoom's video quality is far too low to see individual answers on a camera recording of a screen. Additional cheat protection strategies such as randomizing the exam and versioning the exam (discussed later in this report) can also help mitigate any risk of seeing another student's answer sheet. Also, most "one-room-per-proctor" approaches had students joining Zoom on their phone. Under this policy, any student trying to watch other proctoring feeds would have to look closely at their smartphone, which is noticeable by proctors.

Figure 3.1: Sample video camera setups, demonstrated by course staff. If students use a smartphone to record their workspace and screen, the video quality is too low to see answers, but high enough for proctors to see what the student is doing.

Although the video feed quality is too low to see answers, it is usually high enough for proctors to recognize when a student is visiting disallowed websites or opening disallowed windows, as seen in Figure 3.1. By limiting the number of allowed websites or windows the student can view, proctors can get an idea of what those pages usually look like, and be able to detect when a student visits a different, disallowed page. This can be made easier by adding identifying marks to the pages, such as adding a specific color background to the page, or adding a watermark to the page. However, adding distinguishing features should be balanced with student readability and accessibility: it may not be worth making the exam background bright yellow for proctoring purposes if it also makes the exam difficult to read for students.

In response to schools using Zoom for proctoring, in mid-2021, Zoom introduced focus mode, where participants can only see a video feed of themselves and not other participants, and hosts can see the video feed of all participants. Focus mode also allows each student to share their screen, giving proctors a video feed of each student's screen, while ensuring that students cannot see each other's screens. This would preserve the privacy approach of the "one-room-per-student" approach, while allowing proctors to simultaneously view the video feeds of many students, as in the "one-room-per-proctor" approach. Since the introduction of focus mode, some classes have continued using the "one-room-per-student" approach because both staff and students are accustomed to this approach, while a few classes have started experimenting with focus mode.

Recording

There were two main approaches to collecting proctoring information. In the recording option, students had to make a recording of their proctoring feed and submit it after the exam was over. One common approach to submitting feeds was to ask students to fill out a Google Form linking to their recording after the exam. Staff could then view the proctoring feeds after the exam to detect anomalies.

Setting up the recording may require additional student overhead during the exam, which can contribute to student stress. For example, recording a video feed for several hours can require significant battery power and disk space, which can cause issues if battery power or disk space runs out during the exam. Concerns with disk space can be solved by using Zoom's "record to cloud" option, though this requires a stable Internet connection during the exam.

The second option is live proctoring, where proctors view students' proctoring feeds as the exam is happening. With this option, students no longer need to record their feed. Not recording feeds is better for student privacy, but loses the ability to review recordings after the exam to confirm any suspicions of cheating.

Live proctoring also has the added benefit of a constant TA presence, in case the student has any logistical emergencies. Although many classes had a separate clarifications form for exam content and discouraged students from using the Zoom chat except in emergencies, the ability to contact a TA immediately and receive a response can reduce stress when an incident occurs. Many classes used a combination of these approaches, where proctors periodically viewed live feeds, but students were required to submit recordings for later viewing.

Video and Audio Feeds

Different classes have had different expectations for what should be visible in the video feed during the exam. The first option is to enable screen sharing for a video feed of the student's computer, along with a webcam video feed usually showing their face and possibly some of their workspace. This is relatively low-stress and probably the easiest to set up, since most students have already set up something similar for other meetings such as discussion sections.

The second option is for the student to use their smartphone camera to provide a video feed of their hands and their computer screen. This is slightly harder to set up than the first option, but students generally were able to figure out a working setup. The biggest benefit of this option is that students only need to use Zoom on a single device. For students with slower Internet connections, multiple instances of Zoom is sometimes a concern, and depending on the exact logistics, may also introduce unnecessary stress.

The third option is for the student to use their smartphone camera or webcam to provide a video feed of their workspace, then enable screen sharing on their computer for a video feed of the student's computer. This is the most difficult to set up, because it may require two devices, and filming an entire workspace may be infeasible or unnecessarily complex depending on where students are taking the exam. Additionally, requiring more elements to be in view of the video feed (cheat sheet, scratch paper, hands, pencil, etc.) may cause additional stress, because students may feel the need to periodically check that their video feed is capturing the correct workspace.

Some classes offered pictures of sample setups by asking staff members to take pictures of themselves putting together a proctoring setup. Asking many staff members to follow the setup rules themselves can help ensure that students in different environments would be able to replicate the setup. Also, posting the example setups can help students better understand and replicate the setup. Note that posting stock photos of computer setups (i.e. not photos explicitly taken to replicate an exam setup) may lead to some additional student confusion.

Bathroom Breaks

Different classes had different policies for students who needed to use the bathroom during the exam. Some classes required students to inform a proctor by filling out a form or leaving a note on Zoom. Others simply allowed students to use the bathroom without asking.

Technical Difficulties

Classes needed to plan for technical difficulties during proctoring. The most common technical difficulty encountered was an unstable or broken Internet connection. Other difficulties included real-life inconveniences like a fire alarm or a phone running out of battery, but these all usually resulted in the same loss of proctoring feed.

Most classes were relatively lenient with technical difficulties after the fact, but sometimes required students to submit a report after the exam with a description of what happened. Some policies were also stricter about technical difficulties, requiring students to set up a backup proctoring plan if their main feed stopped working. Regardless of the actual action taken when technical difficulties occur, the communication of potential penalties for an incomplete or broken proctoring feed can contribute to student stress (as described later in the report).

3.6 Communicating Proctoring Policies

The communication of proctoring policies to students can have a significant effect on student stress, perceived effectiveness of the policy, and overall satisfaction with the remote exam. Most classes presented their policies a week or two before the first exam by releasing a logistics document. These documents were usually hosted on Google Docs and ranged from 4 pages to over 15 pages long. Longer policies often caused increased student stress leading up to and during the exam, and several of the longest policies resulted in student complaints. Most classes did not explicitly communicate privacy guarantees to students, which additionally resulted in some concerns about student privacy. This was particularly a concern for classes that collected recordings of students taking the exam. Some best practices for effective communication of proctoring policies are described later in the report.

Chapter 4

Remote Proctoring Case Study: CS 161

4.1 Introduction

In March 2020, the COVID-19 pandemic forced many universities to unexpectedly switch to online instruction [26]. During the pandemic, many classes have had to adjust rapidly to online instruction and adopt new strategies for administering online exams [51].

However, online exams introduce unique challenges for ensuring academic integrity [18]. Students have reported that they are more likely to cheat and that they feel cheating is more acceptable when exams are online, and especially when exams are online and unproctored [15, 40]. These reports are reflected in data that shows higher cheating rates in unproctored online exams [21]. Additionally, students who see their peers cheating may feel that it is rational for them to cheat as well, creating a feedback loop that increases cheating rates [34]. Studies have shown that when given multiple unproctored exams, the cheating rate increases as more students are tempted to cheat and students learn effective cheating strategies [8]. To counter higher cheating rates and preserve exam integrity, we designed and implemented a remote video proctoring strategy.

Our remote proctoring strategy was designed with a few main goals in mind. Proctoring should disincentivize and prevent cheating in honest students, and detect cheating when it occurs. At the same time, it is important that the proctoring logistics should not significantly affect student stress during the exam and preserve students' privacy as much as possible. Finally, proctoring should be accessible and not leave any student at an unfair disadvantage.

During the exam, we asked students to use their smartphone camera to record or broadcast a live video feed of their workspace and computer screen with the Zoom video conferencing software. Additional considerations were made to ensure that our proctoring policy was low-stress and accessible. We also communicated with students and incorporated their feedback to make sure their concerns were heard and accommodated.

We implemented our proctoring policy in a large (>400 students/semester during the academic year, >100 during the summer), upper-division CS class across multiple online semesters during the pandemic. We found that implementing our proctoring strategy successfully decreased cheating rates without increasing student stress levels, creating significant privacy concerns, or making the exam unfair.

4.2 Design Philosophy

Cheating Prevention

Although video proctoring can be used to *detect* cheating, we reasoned that proctoring is most useful for dissuading students *from* cheating. Studies have shown that students are less likely to cheat when remote proctoring is in place, and students report that they are less likely to cheat when they know that they are being monitored during the exam [37]. A well-designed proctoring policy can limit many common, easier methods of cheating on a remote exam, which disincentivizes honest students from trying easier methods of cheating by increasing the probability that they will be caught [14].

To prevent cheating, it is important for students to perceive that the proctoring policy is secure and effective. If students believe that a policy can be easily circumvented or is otherwise insecure, they may be more inclined to cheat.

As a computer security class, we specifically assume that attackers mostly act rationally. A quip from the instructor at the start of the semester was that "If we gave you an exam without proctoring and cheat detection, we'd be insulted if 100% of the class would not cheat," because ethics aside, it would be rational for students to choose to cheat under such circumstances.

This mindset is further exacerbated by students' existing perceptions that it is easier to cheat online, and that cheating online is more acceptable [27]. Thus we valued communicating our policy to students in a way that would convince them that the exam was secure and that neither the student nor their peers would be able to cheat without being detected.

One way we discouraged cheating was by requiring all students to sign an acknowledgment of the university honor code before starting the exam (a practice we also used for in-person exams). Past studies have shown that honor codes change students' perception of cheating and whether their peers cheat [33], though the increased social distance between instructors and students may decrease the effectiveness of honor codes [31]. Future work could compare honor code systems at different universities to see how the honor code affects attitudes regarding cheating on remote exams.

It was also important for us to remember that in-person exams are not completely immune to cheating either. Thus, our goal was not to achieve perfect cheating prevention and detection with our policy, but to approximate the security of an in-person exam while balancing stress, privacy, fairness, and accessibility concerns.

Our proctoring policy was complemented with an exam structure designed to prevent and detect cheating. These anti-cheating measures were designed by considering cheating methods unique to online exams, such as posting questions online, which have become more common during the pandemic [29]. For example, we created exam versions where each student received a randomized variant, included “cheat trap” questions where different variants had subtly different answers, and used the exam platform to record keystrokes for detecting students who copied answers and changed them later. Apart from the randomization, these methods were neither visible nor disclosed to the students, although we did notify students that we also were using unstated methods to detect misconduct.

Student Stress

Exams are already a highly stressful environment for students, especially if the exam is high-stakes, such as a midterm or a final exam in a class that is part of their major. In addition, students are already in a stressful learning environment as a result of the pandemic and the sudden transition to online learning, exacerbating the stress of online exams [16]. Local real-world events throughout 2020 and 2021, such as the seasonal wildfires in California and political unrest in the United States, may also have contributed to increased student stress levels. This combination of stress factors means that policies that may seem reasonable in a normal environment can potentially become very stressful in a remote exam environment.

In addition to preserving students’ well-being, managing student stress is important in designing a fair exam, as students with higher test anxiety will be disproportionately affected by stressful proctoring policies, resulting in lower exam scores that do not reflect their understanding of the material [49].

When designing our policy, we carefully considered every rule and weighed its potential impact on student stress against its effectiveness at preventing or detecting cheating. If a policy had only minimal benefit for preventing cheating but would cause a large increase in student stress, we decided not to implement the policy.

One helpful philosophy for minimizing student stress was remembering that the student’s main focus during the exam should be to finish the exam, and proctoring should only be a second concern. If the proctoring policy was too complex or strict, students may feel like they are being tested more on their understanding of the proctoring logistics, rather than their understanding of the course material.

From other classes’ policies, we found that more complex proctoring logistics correlated with increased student stress levels. In order to minimize the logistics burden on students, we tried to assign the bulk of logistics to course staff wherever possible, so that from the student’s perspective, the proctoring logistics were simple and low-stress.

In addition, we also took care to convey our proctoring policies to students in a way that felt concise, simple, understanding of the circumstances, and accommodating, in hopes of reducing student stress in the weeks leading up to the exam. This involved abstracting away any logistics that the student did not need to know in order to simplify their exam experience and keeping the proctoring instructions for students very short.

Student Privacy

Remote video proctoring is intrusive on students' privacy [23]. In addition to ethics concerns, invasive proctoring is a source of stress and anxiety for students [22]. As instructors, we recognized that we were effectively mandating students to show us video of their home environment and video of themselves in a stressful setting, which in most other settings would be seen as an intrusive and unreasonable request.

Although it may be implied that we were not trying to intentionally violate students' privacy or otherwise be unethical with students' data, it was important to us that we explicitly communicated our intentions to students and made explicit privacy guarantees. We wanted to ensure that all students knew that only their listed proctor would be viewing their video feeds and their video feeds were not being recorded. (If video feeds were recorded, we would have told them how long the recordings would be saved.) Not only do privacy guarantees reassure students that their data will not be used maliciously, but it also communicates to students that we were aware of the intrusive nature of proctoring and that student privacy was an important consideration.

Fairness and Accessibility

Remote exams introduce fairness and accessibility concerns and exacerbate some existing fairness and accessibility issues already present in in-person exams. For example, unlike in-person exams, where every student takes the exam in the same testing environment, students taking remote exams at home could be in different time zones or home situations. When designing a policy, we were careful to minimize the use of any policies that would make exams unfair or inaccessible for all students.

We also recognized that the pandemic and remote learning often disproportionately affected underprivileged students [6]. Rather than leaving special exam accommodations as an afterthought, we worked with instructors experienced with in-person exam accommodations to ensure that students were provided with the necessary accommodations to succeed, whether they were existing accommodations or new policies we devised.

4.3 Proctoring Policy

Zoom

Online proctoring services such as Proctorio, ProctorU, and Honorlock have been criticized during the pandemic for creating stressful exam environments and violating students' privacy [22]. Some of these services use AI to algorithmically detect cheating, which raises additional concerns about biased detection [45]. There have also been reports of students easily circumventing the strict proctoring environments to cheat anyway [19]. These factors, in addition to university policy, contributed to our decision to avoid third-party proctoring software and design a proctoring policy around the Zoom video conferencing tool instead.

One notable benefit of Zoom is that our university was already using it for lectures and sections, so students were more familiar and comfortable with the software.

Student Workflow

For clarity, the instructions for students were arranged chronologically and divided into steps the student should take before, during, and after the exam.

Before the exam, the student should inform instructors of any necessary accommodations. (These are discussed further in the accommodations section.) Students should also download the Zoom video conferencing software on their smartphone and experiment with a video recording setup. Closer to the exam, instructors send out an email with a Zoom meeting link and the name of their exam proctor.

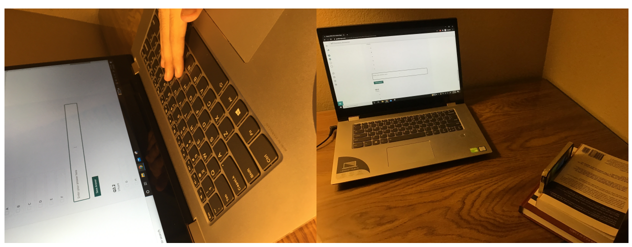


Figure 4.1: Sample video camera setups, demonstrated by course staff. These images were provided to students in the instructions as examples of possible setups.

On the day of the exam, the student should join the provided Zoom meeting. The student's smartphone should be positioned so that their computer screen, workspace, and if possible, their hands are visible in the video feed. Once the student is finished or the exam is over, they can leave the Zoom meeting, and the proctoring is over. Note that in addition to providing a view of both the workspace and the screen, using the student's smartphone for proctoring also prevents students from using their phone to access the Internet or other disallowed resources.

Staff Workflow

First, the instructors in charge of exam logistics collect a list of students and randomly assign each student to a proctor. Each exam proctor checks their list of students and identifies any students who may lead to a conflict of interest, e.g. if a student and proctor are friends outside of class. These students are switched to other proctors.

After confirming their students, each proctor creates a Zoom meeting with the appropriate settings. The Zoom link and proctor name for each student are emailed to students before the exam. (We used a Python script to automate the email process, though any mass-email application would work.)

On the day of the exam, the proctors join their Zoom meetings to monitor their students. During the exam, each proctor takes attendance to note any missing students. Each proctor also writes down any instances of suspicious proctoring feeds, such as a student leaving for an extended period of time, a student using a disallowed resource, or a student's video feed disconnecting. This can all be recorded in one centralized Google Sheets document for easy access. In case of emergency, each proctor also checks the Zoom chat in case any student is trying to contact course staff with a question.

Variations

Our class designed and used a specific proctoring policy, but our policy can be easily adapted to different exam formats with only minor adjustments. Examples of these adjustments are listed below.

Some classes preferred to separate each student in a separate Zoom meeting instead of having multiple students in the same meeting. The student might record this meeting and send the recording to instructors. During the exam, proctors may simultaneously monitor several students' meeting rooms or move between meeting rooms to spot-check students. For this variation, the policy is essentially unchanged from the student's perspective. From the staff's perspective, each student must receive a different Zoom meeting link, possibly with cloud-based recording already enabled, which can be automated with a script. After the exam, proctors may have to review video recordings. We used a similar process in the case of alternate exams where no live proctor was available.

Some classes required different video perspectives from the student. For example, instead of a video feed from their phone showing their workspace and screen, students may be asked to provide a webcam feed showing their face, upper body, and/or workspace. This may require asking students to join the Zoom meeting from their computer instead of their phone.

Some classes required students to additionally provide a screen recording feed during the exam. If multiple students are in the same room, this is only possible by enabling Zoom's focus mode. However, if each student is in their own room, this can be implemented by asking students to join the Zoom meeting on their computer and sharing their screen.

Technical Difficulties

Following our philosophy that the student's main focus during the exam should be working on the exam and not proctoring, we implemented many failsafes in case students encountered technical difficulties. Brief video disconnections were excused. Extended video disconnections were noted, but we would look for additional evidence before making any accusation of cheating. We did not ask students to waste time trying to fix a broken proctoring feed during the exam, in order to let them focus on the exam. Students were allowed to use the bathroom without informing anyone, though proctors could make notes for long or repeated absences, so those exams could be further investigated afterwards.

One important aspect is that any suspicious behavior is not immediately treated as an incident of misconduct. Instead, these cases may result in an oral quiz, presented as a "trust but verify" process, where the student answers a variation of one of the questions they correctly solved on the exam and explains how they come up with the solution. We have yet to have a student fail this oral followup quiz, but its primary intent is as a deterrent: students know that it is very hard to cheat on such an oral quiz.

Accommodations and Alternate Exams

To support disabled students and students who need additional accommodations, in line with university policy, we granted some students 150% or 200% time to work on the exam. These students were proctored by a member of course staff responsible for accommodations during the entire semester in order to ensure student privacy.

To support students in different time zones or with various time conflicts, we released a Google Form for students to request alternate exam times. In light of the circumstances during the pandemic, we accepted every request for alternate exams, though we did ask students to provide a reason before granting requests. We communicated with students over email to settle on an exam time where both the student and a proctor would be available. Because of constraints on staff time during exam writing, the alternate exam questions were the same as the regular exam. To prevent exam leakage from students taking alternate exams, alternate exams were always scheduled after the main exam time. Although students taking an alternate exam might have learned about the exam from a friend, we relied on cheating detection methods and verbal exams to verify that students taking alternate exams were honest.

Communication of Policies

By observing other classes, we found that effective communication of proctoring policies improved students' perception of the policy's effectiveness and reduced their stress.

Before releasing the proctoring instructions, we asked both TAs in the class and TAs from other classes to proofread the instructions and identify any phrasing or policy that could potentially cause issues. We released the instructions to students several weeks before the exam so they would have time to internalize the instructions and ask any clarifying questions.

We presented the instructions to students with a short, human-readable URL that linked to a Markdown page on our course website. All the instructions were centralized on this page, including any updates and revisions, so that students could look in one location and find all of the instructions. We found that this was more presentable and accessible than some other publishing methods such as Google Docs or publishing addendums on Piazza.

The length of the instructions can also affect student perception of the policy. Longer policies tend to appear more complex and stressful, so we aimed to keep our policy exactly one page long. This was significantly shorter than the instructions from other classes in our department, which ranged from 4 to over 17 pages. This was achieved by hiding any unnecessary logistics from the student and leaving out justifications for the policy, which students do not need to remember for the exam.

To discuss the justification behind the policy, we instead used other channels such as a 10-minute time slot in a lecture, a Piazza thread for questions, a feedback form, and a short check question on homework. This allowed us to communicate our considerations for exam security and student well-being, and it allowed students to offer constructive feedback or ask any clarifying questions.

We also adjusted the tone of the instructions to make them accommodating and welcoming to students. Notably, we avoided any language that would sound threatening to the students, such as saying that failure to follow a proctoring rule could result in a score of 0 on the exam or an accusation of cheating. We felt that such threats create a more stressful proctoring experience and give the impression that we are testing students on their knowledge of the proctoring rules rather than knowledge of course content.

We also reiterated that we assumed that students were acting in good faith and were not trying to cheat. Even though our policies were designed to prevent and detect cheating, we did not want to antagonize students by treating them as malicious actors. To emphasize this, we offered reassurances that there were failsafes in place for honest students who might accidentally make a proctoring mistake, such as the "trust but verify" oral exam process.

Privacy Policy

In our instructions to students, we explicitly communicated a set of privacy guarantees in order to reassure students that we would not be acting unethically with their video feeds. We specified that course staff would not save any images or recordings from the proctoring session after the exam is over, and that course staff will respect student privacy and not disclose any information from the proctoring session after the exam, except in cases of academic dishonesty. This guarantees that no data from proctoring would leak outside of course staff. In the live proctoring setup, since no recordings are made, any video or image of students' feeds is gone after the exam is over, and only text records of suspicious exams remain. In the recorded setup, the policy could be modified to specify that all proctoring recordings will be deleted after the exam is over.

We allowed students to join the Zoom meeting with their student ID as their display name for anonymity to other students and the instructors. This allows honest students' identities to be obfuscated to proctors and other students, who will not know which student the video feed belongs to. However, if the proctor notes the student being dishonest, when investigating academic dishonesty, the student can be de-anonymized by looking up the student ID.

Although we asked proctors to swap out any students they might have a conflict of interest with, we also gave students the opportunity to switch proctors with no questions asked.

To protect students against false positives and accusations of cheating, we promised that every case of academic dishonesty would be manually reviewed by an instructor, students would have an opportunity to discuss the situation with an instructor, and the first step of the manual review process would involve the trust-but-verify oral quiz.

To ensure that these guarantees were honored, we gave all proctors a 10-minute briefing describing the privacy boundaries and the harsh consequences for violating students' privacy.

4.4 Analysis of Effectiveness

A student survey collected by the department in the Fall 2020 semester asked students to rate the remote proctoring policies of each class in three categories: clarity of instructions, perceived effectiveness, and stressfulness. In general, the results showed a positive correlation between perceived effectiveness and student stress. Classes with minimal proctoring or no proctoring at all were rated as ineffective but stress-free, while classes with strict proctoring were rated as effective but high-stress. However, our proctoring policy was the only one rated as both low-stress and effective by students, as shown in Figures 4.2 and 4.3.

In addition to having one of the highest effectiveness ratings and one of the lowest stressfulness ratings, our class had the highest rating in clarity of instructions, as shown in Figure 4.4. This suggests that our strategies for clear communication to students were successful.

Our proctoring policy also appears to have decreased the cheating rate compared to unproctored exams. When we applied the same cheating detection methods on unproctored exams and exams proctored with our policy, we found a decrease in cheating cases detected. However, it is possible that students found different cheating methods to circumvent our proctoring method that these detection methods did not catch.

We compared our class to another similar class, which used a similar exam platform and the same cheating prevention and detection methods (e.g. randomization and cheat-trap detection), but did not proctor exams. In a class of 330 students, the unproctored class reported 19 cases of misconduct on the midterm and an additional 7 cases on the final. In a class of 450 students, our class reported 2 cases of misconduct on the midterm and an additional 1 case on the final. Although just an anecdotal data point, this does suggest that the presence of proctoring reduces some common forms of cheating.

4.5 Conclusion

With careful policy design and communication with students, it is possible to design a remote proctoring policy that preserves the integrity of exams without overstressing students, violating student privacy, or creating an unfair exam. Later in this report, we investigate how these policies were further adapted to support hybrid exams, where part of the class takes the exam in-person, and the other part of the class takes the exam remotely.

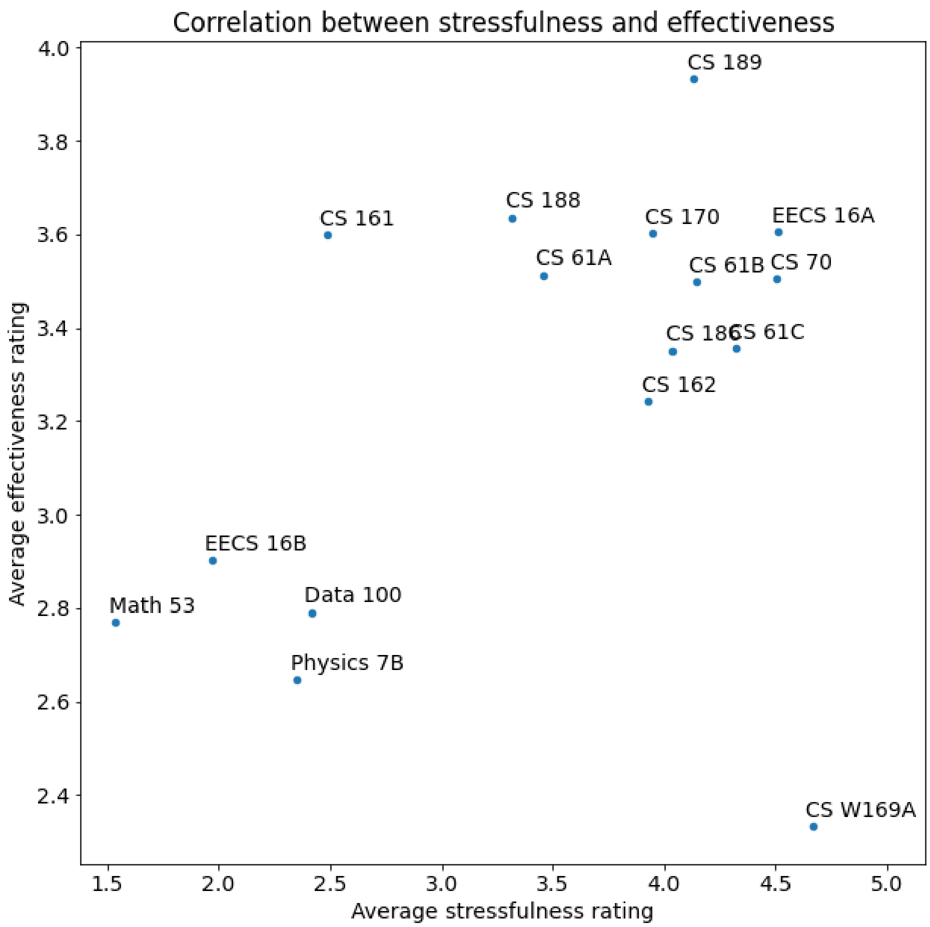


Figure 4.2: Plot of student feedback from a Fall 2020 survey of exam proctoring policies. In general, policies with better perceived effectiveness were rated as more stressful, but our class was an exception to the trend.

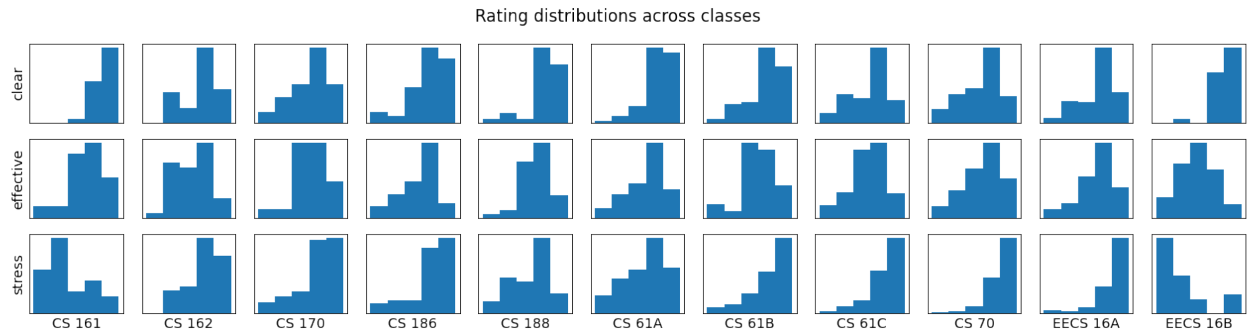


Figure 4.3: Plot of student feedback from a Fall 2020 survey of exam proctoring policies. The rating distributions across classes usually showed either high-stress proctoring with good perceived security, or low-stress proctoring with poor perceived security. Our class, however, showed low-stress proctoring with good perceived security. Additionally, our policies had the highest ratings for clear communication.

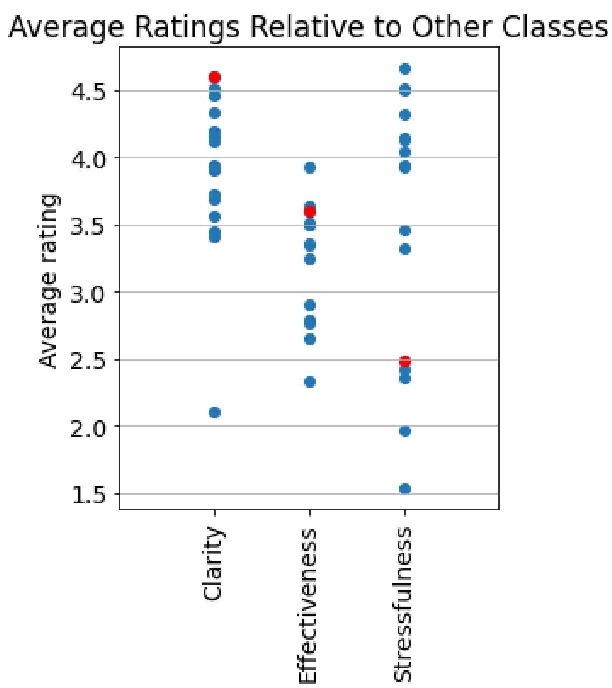


Figure 4.4: Plot of student feedback from a Fall 2020 survey of exam proctoring policies. In all three metrics, CS 161 (red dot) performed well compared to other classes. Most other classes performed poorly in at least one of the metrics. (Note that for clarity and effectiveness, higher is better, and for stressfulness, lower is better.)

Chapter 5

Remote Proctoring Best Practices

This section is from a report titled "Developing an Effective Remote Proctoring Policy" that we wrote with input from students, TAs, and instructors in the EECS department. It was distributed to instructors in the EECS department throughout the pandemic as a guide to balancing exam security and student stress in a remote proctoring policy. Outside of the EECS department, the university, in collaboration with the EECS department chairs, organized several meetings with faculty members from other departments to share these best practices from the EECS remote proctoring pilot program, and the findings in this report were summarized and distributed to instructors for consideration when developing their own remote exams.

Zoom's relatively extensive functionality makes it tempting to devise creative, often highly complex proctoring logistics designed to eliminate every possible form of cheating. However, the most important thing to remember when developing an effective remote proctoring policy is that **exams are already a highly stressful environment for students**.

Additionally, we need to be aware of the fact that **we are not in a normal learning environment**. Student stress levels are inherently higher as a result of the remote learning environment and the many real-world situations they are facing (e.g. COVID-19, California wildfires, political unrest). Thus, it is unreasonable to try and recreate the experience of an in-person exam. For example, we need to remember that everyone's home environment is not comparable to a lecture hall or classroom where in-person exams are administered. We need to remember that students may now be in different time zones. Students are much more likely now to encounter unexpected last-minute emergencies, such as a student or family member getting COVID. All of these considerations, which are normally less relevant or entirely irrelevant for in-person exams, are now very important when developing an effective remote proctoring strategy. Instructors should recognize, accommodate and plan around these contingencies, rather than ignore them entirely and try to simulate an in-person exam.

Although the balance between reducing student stress and effectively preventing cheating is ultimately up to the instructor, our experience is that there is a non-linear relationship between student stress and effectiveness at preventing cheating. In other words, the large amount of additional stress resulting from stricter policies is not worth the minimal amount of additional cheating prevention from those policies. Students' perception of exam security also follows a similar trend: a small increase in perceived effectiveness of a proctoring policy is correlated with a large increase in stress caused by the policy. This tradeoff should be carefully considered when developing any set of remote proctoring policies.

A well-designed and effectively communicated proctoring strategy can optimize for both parts of this tradeoff. CS 161's proctoring policy was the only one in the EECS department rated as both "less stressful" and "effective at preventing cheating" by students. The rest of this section describes the considerations we made and the way we communicated our policies to achieve this effect for students while maintaining exam security comparable to that of an in-person exam.

This past semester, we have seen many examples of policies that increase student stress without proportionate additional cheating detection. For example, some policies require the student to hold their student ID to the camera for 10 seconds and state their legal name out loud. The reasoning for this policy is that some students might have others take the exam on their behalf. However, we believe the additional stress incurred by adding extra requirements before the exam is not worth catching the extremely small number of cheaters who would invite someone else to take the entire proctored exam for them.

This policy also highlights two more important points to consider when developing remote proctoring policies. First, a student in a class with this requirement pointed out that they go by a different name from their legal name, and saying their deadname out loud causes dysphoria and discomfort. Any analysis of a potential proctoring policy should always **consider ways in which vulnerable or marginalized groups are affected by the policy**. Although a policy like "state your legal name aloud" may seem innocuous, this may not be true for everyone in our large, diverse student body. We especially do not want to make specific groups of students feel disproportionately stressed or uncomfortable as a result of any policy.

A simple alternative to the "show your ID and state your legal name aloud" policy is to use the roster of student ID photos provided by CalCentral and referencing those when reviewing recordings or live video feeds. This has the same end result of catching anybody taking the exam on behalf of someone else, but requires no additional effort on the student side. Our philosophy when writing proctoring strategies is that **when possible, let the instructor do the work instead of the student**. The student is in a much more stressful position taking the exam than the instructor proctoring the exam, so when the choice is available, additional logistics should always be handled on the instructor side. This minimizes the amount of logistics students need to keep track of, which decreases their overall stress levels during the exam.

This philosophy is related to another important idea when writing proctoring policies: **expose the minimal set of necessary logistics to the student**. While cheating detection strategies can be arbitrarily complex on the instructor end (as TA resources allow), we should aim to abstract as much of the logistics away as possible so students see a simpler, and thus less stressful, version of the proctoring policy. CS 162 had a great success story this semester with exposing less logistics to students: their Midterm 1 required students to set up a Zoom meeting with very specific settings, which led to student complaints and higher stress levels. However, for subsequent midterms, they developed a way to automatically create Zoom meetings with the correct settings for students, which helped decrease stress levels significantly.

When writing a policy document for students to read, **tone is just as important as the actual policies**. Ultimately, most Zoom proctoring policies developed this semester had similar logistics and setup requirements, and much of the variance in stress levels came from the way those policies were presented. If policies are presented in a way that is accommodating and welcoming to students, then students will be more receptive to working with those policies. Conversely, if policies are presented in a way that is antagonistic to students, then students are more likely to complain and protest, because they (often rightfully) conclude that the policies were not written in good faith with their stress levels in mind.

Student-facing proctoring instructions should be as short and concise as possible. The CS 161 student proctoring instruction doc is exactly one page long; all other documents we've seen are at least 4 pages long, with some stretching as long as 17 pages. In our opinion, there is no reason for a proctoring doc to be longer than one page, single spaced, 12pt font. Additional details and edge cases can be discussed during lecture (CS 161 devoted 10 minutes of a lecture to discussing logistics and answering questions), or on Piazza, but should not be part of the main instructions. Longer instructions give off the impression that we are testing students on their knowledge of the proctoring logistics, rather than their knowledge of the course material. Additionally, short instructions does not necessarily imply dense instructions: one metric CS 161 used when writing instructions was to ensure that a student would be able to read over the instructions once and understand what to do.

The presentation of instructions also matters. We've seen most classes use Google Docs, which, while easy to edit collaboratively, is not always the best way to concisely present information. The link is difficult to find, and there is a tendency to be wordier when given the freedom of a long document. Some classes have also released addendums to policies scattered across Piazza posts, which adds unnecessary confusion and logistical overhead for students. CS 161's exam logistics are centralized in a single place and located at a short, easy-to-remember link (cs161.org/exam). The instructions are organized into concise bullet points and sections that clearly describe the proctoring workflow chronologically, with edge cases such as technical troubleshooting located in a section at the end of the document. We do not use the instructions page to discuss the philosophy behind our policies or the purpose of each rule, because these are not relevant for the student when they are preparing for the exam. This discussion is still important to show students that we are thinking of their stress and privacy, but it should happen elsewhere. The exam instructions page is only for clear, short instructions describing exactly what the student must do to proctor their exam.

Proctoring instructions should avoid threatening students. The most common threat that appears in proctoring instructions is that students will receive a 0 on the exam for not following a particular rule. Again, this creates the impression that students are being tested on their knowledge of the proctoring rules and not the actual course content. We don't believe that threats are necessary to force students to follow the proctoring rules: in

CS 161, all students followed the instructions to the best of their ability without any threats as forcing functions. Furthermore, when making such threats, the cost of a false positive is often much greater than a false negative. There have been several cases this semester of honest students accidentally breaking a proctoring rule, causing severe anxiety about the possibility of failing the exam and the class or being referred for academic dishonesty. On the other hand, a cheating student breaking the proctoring rules will usually get caught using other detection methods (see below) and receive a penalty anyway, so the penalty for breaking the proctoring rule is unnecessary for them. It is also unclear whether these threats do anything to dissuade students intent on breaking the proctoring rules and/or attempting to cheat. Certainly, if a student is determined to cheat, then yet another threat of receiving a 0 on the assignment for breaking proctoring policy would not deter them, because cheating already has the same consequence.

Proctoring instructions should always assume students are honest and acting in good faith. Although proctoring is implemented to prevent cheating, we should never automatically assume that students are dishonest and malicious. Explicitly stating that the instructors believe that students are being honest gives students reassurance that we are working with them, not against them. In the stressful environment of a remote exam, it is easy to make an honest mistake. There should be failsafes in place to protect honest students who run into technological problems or make accidental mistakes, and these should be clearly communicated in the instructions.

The student's main focus during the exam should be to finish the exam, and proctoring should only be a secondary concern. This is something we explicitly state in our instructions. Even if an honest student does everything right, there are many ways for technology to fail during the proctoring session, and they should not be penalized or additionally stressed as a result of non-malicious tech failures. As a failsafe, we allow students who run into significant tech issues to finish the exam unproctored and take a verbal exam afterwards, as this is a less stressful and fairer (to the honest student) alternative than wasting time trying to troubleshoot tech during the exam.

Proofread and focus-group proctoring instructions before publishing them. Many cases of proctoring doc disasters this semester could have been easily avoided with a few rounds of proofreading and focus grouping. The EECS teaching community is very large, and many of us are willing to read over proctoring docs and give feedback before they are released. Students should not be guinea pigs fixing mistakes and catching inconsistent edge cases in proctoring instructions. The CS 161 instructions went through dozens of revisions before being released, including fixing typos, adjusting inconsistent rules, and correcting for tone.

Release proctoring instructions early. Proctoring instructions should not be released the week of the midterm, when student stress level is already highest. This effect is compounded with long, dense proctoring docs, because it creates the perception that we are giving students last-minute material to study just before the exam. We recommend a 2-week buffer at minimum (i.e. instructions should be released 14 days before exam day). CS 161 released logistics before the beginning of the semester in fall and the second week of class in summer (equivalent to a 4-week buffer in a regular semester). However, note that releasing polished instructions later is still better than releasing messy, error-laden instructions earlier.

Students are entitled to privacy guarantees. Remote proctoring is inherently intrusive and invasive on students' privacy. Even though as instructors, it may be clear to us that we are being honest and ethical, it is important to explicitly communicate to students exactly what we will and won't do. This also has the added benefit of communicating to students that we care about their privacy and well-being: even students who don't have privacy as their highest priority may feel better when seeing that student privacy was an important consideration.

Students should know exactly who will and won't be viewing their proctoring live feeds or recordings. The list of people viewing their feeds can be as broad as "all course staff" or limited to specific TAs (which is what CS 161 does). Students should also know who won't be seeing their proctoring feeds: although it is clear to us that we won't be publishing video recordings to the world or sending them to a third-party company, this should be explicitly communicated to students, because it demonstrates that we have thought about their privacy. Giving students a list of who will and won't be proctoring them also allows them to consent to proctoring. CS 161 had a few cases of students indicating they were uncomfortable with one of the TAs on course staff, so we gave them a guarantee that the TA they're uncomfortable with would not be seeing their proctoring feed.

Students should know if their video feeds are being stored, and if so, how long they are stored. Although it is implicit that we won't be keeping video feeds forever, again, it is important to communicate this explicitly to students to give them peace of mind and demonstrate that we are thinking of their privacy. For example, CS 161 does not store any recordings, deletes all video after the exam is over, and forbids proctors from taking screenshots or video recordings of students. This is especially important when proctors know students in their class (see below).

To enforce our privacy guarantees, CS 161 **briefs all proctors on course staff about the importance of respecting students' privacy and boundaries**. Although most of these rules are common sense (e.g. don't store recordings and share them outside of course staff), talking about them explicitly in a briefing is a good way to remind proctors that they should be thinking about student privacy as they proctor the exam. Because this is on the proctor side and not the student side, we can be much more strict about these rules and enforce consequences for breaking them, all without imposing additional stress on students. In fact, a stricter privacy policy is beneficial for students, because it reduces anxiety related to filming themselves in a high-stress environment for an extended period of time. It is reassuring for students to know that their recording will be deleted, will not be sent to third parties outside of course staff, will not be shared publicly, etc.

Because students are largely being proctored by their peers (undergraduate TAs), the privacy of proctoring feeds is an especially big concern. There have already been several cases of proctors discussing the contents of video feeds with other students outside of course staff without the consent of the student being proctored. While it is impossible to prevent this entirely, we should make it clear to proctors that this is a clear violation of student privacy, and that there are serious consequences associated with violating student privacy. Just as we would not allow a TA to publicly talk about what score an individual student gets on an exam, we should also not allow TAs to publicly talk about anything they see or hear during exam proctoring. Also, as mentioned, students are often uncomfortable being proctored by one of their peers (compared to being proctored by a faculty member).

CS 161 reduces the risks associated with peer proctoring by informing students of who will be proctoring their exam, and giving them the opportunity to change to a different proctor if they feel uncomfortable with their assigned proctor (no questions asked). Before the exam, we ask proctors to review the list of students they will be proctoring and identify anyone they know personally or otherwise have a conflict of interest with. We then transfer these students to another proctor before informing students of who their proctor will be. This reduces the potential for conflicts of interest or unconscious biases from undergraduate proctors during the exam (e.g. a proctor watching a friend's video proctoring feed more attentively than others).

While this is not true for the "one-room-per-student" approach (see below), CS 161's "one-room-per-proctor" approach lets students see other students' video feeds during the exam. This represents an exam security issue (discussed below), but it also represents a student privacy issue. Although the vast majority of students will not be paying close attention to others' video feeds during the exam, we took some precautions to let students preserve their privacy. We allowed students to join the proctoring session with their SID instead of their name to anonymize them to other students and their proctor. Also, we allowed students to opt for the "one-room-per-student" approach if they felt uncomfortable showing their proctoring feed to other students. A small handful of students opted for these alternatives.

When analyzing the effectiveness of a set of remote proctoring policies, it helps to compare them to the effectiveness of our in-person proctoring policies. While it may be tempting to aim for 100% detection and 0% cheating with remote proctoring, it is important to put things in perspective and remember that **there are many methods of cheating, both detectable and undetectable, in in-person exams as well. Our goal was not to achieve perfect exam security, but to approximate the security of an in-person exam while keeping student stress levels at a minimum**.

In an in-person exam, we typically require students to hand over their smartphone while they use the bathroom. This prevents someone from using their phone from looking up answers in the bathroom, but it does not stop someone with two devices or a friend waiting outside. To replicate this security for the online exam, we ask that students leave their smartphone recording their screen if they need to leave and use the bathroom. This also prevents someone from looking up answers on their phone, but does not stop someone with a third device.

While it may be tempting to add a policy to prevent this, we have not seen any policy successfully defend against this without serious concerns about privacy and student stress. Instead, we accept this vulnerability as being comparable to an in-person exam and rely on better detection methods unique to online exams, as described in the next paragraph.

For detection, we rely on proctors to make note of when students leave for the bathroom. We do not ask students to announce or note that they're using the bathroom, because we try to let proctors do the logistical work instead of the student when possible. Then, we can cross-reference keylogs of their exam from before and after the bathroom session to see if their submission changed unexpectedly as a result of the bathroom break. This is an example of a remote exam cheating detection policy that improves on in-person detection strategies without increasing student stress.

This type of exam security analysis (bathroom devices, peeking at neighbors, video feed quality) was discussed with students at the start of a lecture a few weeks before the midterm, in order to be transparent about our policies and why we decided to implement them. We also solicited student feedback for ways to improve the proctoring policy: although we ultimately didn't adopt any suggestions, the open feedback channel communicated to students that we were willing to listen and adjust, and ultimately made students feel like we were working with students, not against them.

The midterm is 120 minutes, and the final is 180 minutes. However, we advertise the exams as being 110 and 170 minutes long, respectively, with the final 10 minutes being a grace period for distractions given to everybody. This blanket accommodation accounts for students who encounter brief distractions, use the bathroom, or want to print out a copy of the exam after it starts. Although our policy of giving every student a fixed time limit didn't change, the modified wording of a 10-minute grace period changes students' perceptions and lets them feel less stressed about losing time from distractions.

Students with major or prolonged distractions (e.g. fire alarm, noisy environment, Internet outage) can reach out to course staff before or during the exam, and we offer time extensions on a case-by-case basis to offset prolonged distractions. Very few students have requested time extensions, but those who have usually have legitimate reasons.

Because 22.4% of our students indicated that they did not have quiet, distraction-free environments for the exam, we allowed headphones and music during the exam. We also did not require students to turn on their microphones and send us their audio feed during the exam (out of necessity because of the "one-room-per-proctor" setup). While this poses a security risk of getting answers through audio, we felt that giving as many people as possible the opportunity to take the exam fairly in a quiet, distraction-free setting was more important. Ultimately, the goal of proctoring is to create a fairer exam environment by reducing cheating. **Proctoring policies that increase inequity defeat the purpose of exam proctoring**.

Also, when communicating with someone else through audio, the proctor will usually notice the student talking, which lets us investigate for further cheating evidence. The "one-room-per-student" setup can probably safely record audio without adding too much student stress, but we recommend allowing headphones and music in order to ensure fairer, more consistent exam environments for honest students.

Chapter 6

Hybrid Exams

6.1 Timeline of Hybrid Exams

On January 11, 2021, UC Berkeley made the announcement that classes in the Fall 2021 semester would be primarily in-person, with further details to be determined later [11]. On March 16, 2021, an additional announcement was made stating that classes with enrollments of 200 or more students would be offered remotely. Discussion sections and labs would be held in-person, though some discussion sections would be offered remotely. This announcement also noted that remote options were being evaluated for international students unable to return to the United States and students with medical conditions [12]. On August 31, 2021, after the start of the semester, it was confirmed that international students and students with disabilities were the main reason behind keeping remote instruction for larger classes [25]. The enrollment threshold was set at 200 because it was the maximum gathering size permitted by public health officials, even though that limit was removed by the time instruction started [42].

In August 2021, UC Berkeley returned to mostly in-person instruction for the Fall 2021 semester as planned, and initial plans to start the semester with one week of remote classes were called off. Despite the university's plans, many students in August 2021 were still unable to or uncomfortable with attending classes and taking exams in-person. Also, ongoing developments in the pandemic, including the spread of more dangerous variants, required flexibility and support for remote students beyond what was in university policy, especially for large classes like EECS classes.

6.2 Motivation for Hybrid Exams

In mid-August 2021, just before the semester started, a group of TAs from various courses got together to discuss the motivation behind offering remote exams to all students, not just students with a remote exemption from the university. One important point stressed during this meeting was that the pandemic was still ongoing, and students might not be comfortable taking an exam in a crowded room with many other students. A student who gets sick but is forced to take the exam in person might pose a health risk to other students and staff. Additionally, remote exams have accessibility benefits that extend beyond the pandemic (discussed later in this report), and one goal of developing a robust hybrid exam format was to keep those benefits even after the pandemic ends and most students return to in-person instruction. Finally, the pandemic situation was still very uncertain at the start of the fall semester, especially with the Delta variant of COVID-19 threatening to force classes back to remote instruction [52]. TAs emphasized the need for flexibility when exams could be fully in-person, fully remote, or anywhere in between.

6.3 Overview of Hybrid Exam Policies

The level of support for remote students during the fall semester differed among classes, and these differences extended to exam formats. Some classes allowed students to take the entire class remotely, including exams, with no questions asked. Other classes required students to present documentation proving that they had a need for remote exams, such as not physically being in Berkeley, or medical reasons.

Note that some classes opted to continue fully-remote exams during the hybrid semester, even as other parts of the class (e.g. discussion sections and office hours) returned partially or fully in-person. This allowed classes to adopt the benefits of remote exams, such as providing a fallback for students who get sick on exam day, without the additional logistical overhead associated with providing both in-person and remote options. A few classes instead enforced only in-person exams during the hybrid semester, which effectively restricted enrollment in the class to students who could physically be present for exams.

6.4 Synchronous Hybrid Exams

Almost all classes adopted a synchronous format for hybrid exams, which meant that most students took the exam in-person using the same traditional pre-pandemic format (on paper), while remote students followed the synchronous online exam format discussed earlier. This was the most practical choice for hybrid exams because synchronous exams most closely approximate in-person exams, and synchronous remote exams could reuse infrastructure developed during the fully-remote semesters that was familiar to staff and students by now.

In-Person vs. Remote Exams

One additional logistical challenge posed by hybrid exams is determining which students are taking the exam in-person and which students are taking the exam remotely. Classes that request justification or documentation before granting hybrid exams also have to decide when to allow a remote exam.

In the days or weeks leading up to the exam, students need to indicate which format they were using to take the exam. Most classes simplified this task by defaulting students to the either the in-person or remote option, asking students to specifically fill out a form or reach out to instructors to request an exam format different from the default.

The majority of classes defaulted students to the in-person exam option, with the remote option available for students who needed it or requested it (depending on class policy). This was consistent with the consensus that in-person exams provide better exam security and fewer logistics issues than remote exams. However, the classes that offered fully remote exams could be considered as defaulting students to the remote option, with limited in-person availability for a few students who needed to take the exam in-person.

Flexibility in granting remote exams was especially important during the Fall 2021 semester, when developments in the pandemic had the potential to rapidly shift public health and university guidelines. In the most widespread example, students could get sick or get exposed to someone with COVID-19 in the days before an exam, necessitating last-minute switches to remote exams. Although it has not happened on a university-wide scale yet, it is still possible that the pandemic suddenly forces classes to become remote again, which would require all students to switch to the remote exam. Non-pandemic-related events, such as potential safety threats on campus or air quality issues from California wildfires, may also necessitate sudden switches to remote exams.

Exam Content

Although online exams tried out many new structures to reduce cheating and leverage the computer-based exam format, hybrid exams largely returned to traditional in-person exam content, as discussed earlier. This was necessary to ensure parity between students taking the exam in-person and students taking the exam online. For example, a coding question that asks for fully compiling code and is autograded by executing the code may work in a remote exam setting where all students have access to a compiler, but would not work in a hybrid exam, where in-person students cannot use a compiler.

Another important consideration for hybrid exam content is that in-person students are handwriting their answers on a printed exam. If remote students typed out their answers, questions had to be designed such that typing would not provide an unfair advantage over handwriting. For example, a coding question that requires a line of code to be repeated often may give an advantage to remote students who can copy and paste text. In the other direction, a question whose solution uses diagrams or mathematical symbols may give an advantage to in-person students who can more easily draw diagrams or write symbols in their answer.

The other approach is to have remote students handwrite their answers on paper and submit scans or photos of their answers at the end of the exam. However, this format also requires careful exam design: if a question requires annotating a diagram on the exam, or annotating the question is helpful for solving the question, then remote students who are unable to print the exam and are writing answers on a blank sheet of paper may be at a disadvantage.

Remote students who receive a digital copy of the exam also have the benefit of viewing the exam more flexibly. For example, a student taking the exam remotely could open a PDF copy of the exam in two windows and get a split-screen view of any two pages on the exam, while a student taking the exam in-person is limited to using one stapled packet of the exam, which makes it more difficult to view multiple pages of the exam simultaneously, especially if the pages are not adjacent. Although this was considered by some classes when developing hybrid exams, it is unclear whether this provides a statistically significant advantage for remote students.

Allowed Resources

As with exam content, the limits on allowed resources mostly returned to traditional in-person policies, since in-person students cannot access the Internet or computer-based resources such as IDEs. However, because enforcing the number of cheat sheets is difficult in a remote exam, some classes increased the amount of allowed cheat sheets in-person so that remote students would not have an incentive to cheat and gain an unfair advantage by bringing additional cheat sheets.

Clarifications

Processing in-person and remote clarifications at scale poses several logistical challenges that are not present when processing exclusively in-person or exclusively remote clarifications. Clarification requests are now coming from at least two different sources, creating two queues that must both be processed in time. Also, published clarifications may be relevant to only in-person or remote students, such as when a typo is caught on only the printed exam, but not the remote exam. This creates additional overhead, both in checking two different exam formats, and in wording clarifications carefully to avoid extra confusion from the differing exam formats.

As discussed earlier, the barrier to entry for clarifications is lower for an online exam. This introduces some fairness concerns where students taking the exam remotely submitted more clarification requests. However, it is unclear whether being able to submit more clarification requests translates into an overall advantage, especially if questions are only answered in clarifications published to the entire class.

Timing Accommodations

During hybrid semesters, a large majority of students returned to campus, so hybrid exams required fewer accommodations for alternate-time exams than fully online exams during the pandemic. However, many classes still offered alternate exam times, often directly after the standard exam time, for reasons outlined earlier in the in-person exam section.

DSP Accommodations

Accommodations for both in-person and remote exams need to be considered during hybrid exams. Both in-person and remote exams may have timing-related accommodations, such as giving 150% or 200% time to complete an exam. Scheduling all the possible combinations of alternate exam times, extended exam times, and remote exams can pose logistical challenges that require manually processing more unusual exam times where only one or two people are taking the exam.

Although setting up hybrid exams requires more logistics work, having the option of remote exams for students who need them provides significant accessibility benefits. For example, students who need a digital copy of the exam for accessibility reasons can simply be offered a remote exam. Also, if a student needs to take the exam at a significantly different time from other students (e.g. the exam is scheduled for the evening, but the student cannot take exams later than 5:00 PM), an asynchronous proctoring setup can be used to proctor this student without allocating additional staff hours. This is discussed in more detail in the accessibility section of the report.

6.5 Synchronous Computer Exams

One possible variation on the hybrid exam format is to have students who choose the in-person option take the exam on a computer in-person instead of on paper. This could be achieved in a computer lab (possibly with specialized computers for online tests), or in a regular classroom or lecture hall where students bring their own computers. In-person computer exams quickly encounter logistical issues when scaling to very large classes: specialized computer labs cannot fit entire classes with hundreds or thousands of students, and a lecture hall with thousands of students taking a computer exam requires infrastructure to ensure that computers have consistent power and Internet access during the entire exam. These logistical issues were the main reasons synchronous computer exams were not tried at Berkeley during the pandemic. A specialized computer lab for online tests may offer a scalable solution [53], but UC Berkeley currently does not have such a facility or proctors to manage it. Another possible approach that may be investigated in future semesters is to use question generators to give each student a different version of the exam and spread out the exam times over several days, though this may encounter potential exam security issues. Also, most of Berkeley's lecture halls are old and, even when recently refurbished, often lack enough power outlets at the seats to provide a power source to every student. Exams can last two or three hours, which easily exceeds the battery life of some students' computers.

However, if the logistical issues are worked out, computer exams for all students provides a simple way to ensure parity between students taking the exam in-person and remotely. Furthermore, with a mix of in-person and remote proctoring policies, exam security for in-person students could be comparable to that of paper exams. Further work could investigate the feasibility and benefits of scaling digital exams to large classes.

Chapter 7

Hybrid Exam Case Study: CS 61C and CS 161

7.1 Design Philosophy

In the Fall 2021 semester, CS 61C (lower-division computer architecture) and CS 161 (upper-division computer security) took similar approaches to hybrid exams, helped in part by both classes sharing an instructor, several TAs working for both classes, and active collaboration between staff from both courses.

The overarching design philosophy for these classes was to fully support both in-person and remote options, allowing students to self-select whether they were comfortable with in-person instruction or whether they preferred to take the class remotely. With the pandemic still ongoing, there were many reasons why students might need to take a remote exam. Students could be immunocompromised, unvaccinated, sick, or at greater health risk in general if they took an exam in-person, and forcing students to choose between their health and the exam seemed like a health hazard to other students, in addition to a source of stress for students. By fully supporting remote options, both classes were able to encourage students to stay home and take exams remotely if they were showing symptoms of COVID-19 or otherwise felt unwell. Although remote exams might facilitate more cheating than in-person exams, these classes chose to prioritize student health and safety first, relying on remote proctoring developed in earlier semesters to preserve exam integrity.

7.2 Communication of Policies

At the start of the semester, both classes communicated their support for remote options in the first lecture and in the course policies page. Both classes also sent out an introductory form to students, surveying student preferences and communicating some initial information about remote exams. In the form, students were asked about what time zone they were in during the semester. In contrast to the breakdown seen earlier in the report during fully-remote semesters, almost all students reported being in the same time zone as the university, as shown in Figure 7.1. However, some students reported that they would not regularly be on campus throughout the semester, even if they were in the same time zone, as shown in Figure 7.2.

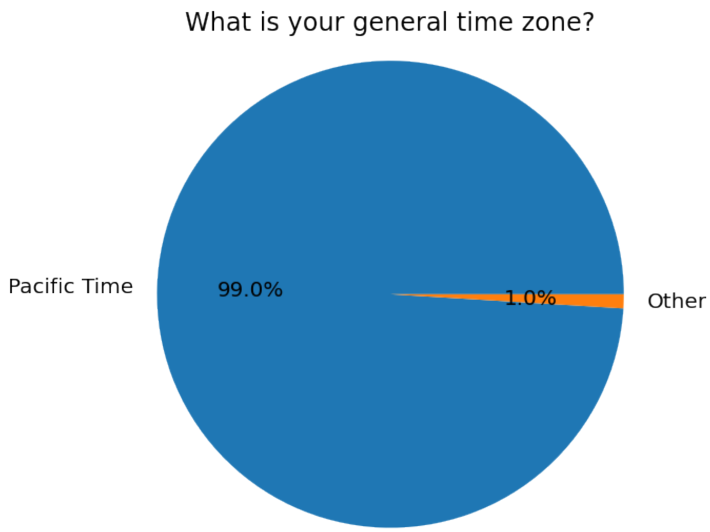


Figure 7.1: Percentage of CS 161 and CS 61C students (upper-division computer security class of 450 students, and lower-division computer architecture class of 1000 students) in the same time zone as the university during Fall 2021, a hybrid semester.

Later in the form, students were informed that although remote exams would be allowed for any valid reason, the default option for students would be in-person exams, and students had to contact staff to request a remote exam option. Students were asked to opt-in to default in-person exams by acknowledging the following statement: "I understand that I will be assigned to an in-person exam by default and that I will later have the option to opt for a remote exam." This simplified processing requests to change exam formats, under the assumption that most students would be taking the exam in-person at the standard time. If developments in the pandemic forced classes to become mostly or fully remote, a contingency plan was in place to switch exams to be fully-remote, discarding any default option that was previously accepted.

Although students were offered an option to contact instructors if they were unsatisfied with opting into a default in-person exam, neither class received any concerns from students, suggesting that this initial policy works well even at large class sizes.

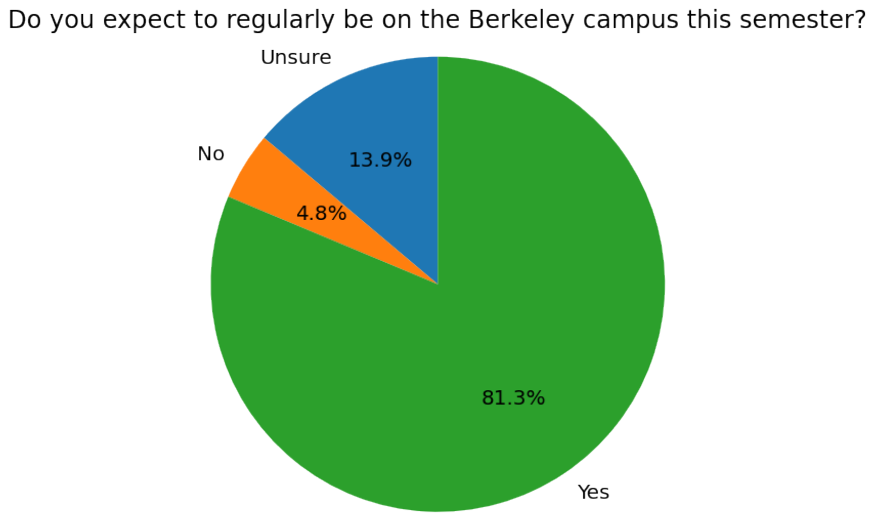


Figure 7.2: Percentage of CS 161 and CS 61C students who reported they would be on campus regularly during Fall 2021, a hybrid semester.

7.3 Student Workflow

To simplify student-facing logistics, all requests to change exam format from the default format went through a single, centralized Google form, linked on the course webpage. This included requests for a remote exam, alternate exam times, and any additional accommodations not already covered by a student's preexisting DSP accommodation.

The form starts by asking if the student prefers an in-person or online exam. If the student selects an in-person exam, the form asks if they have any in-person preferences, such as a left-handed desk, a seat near the front of the room, or a specific room on campus closer to home.

If a student selects an online exam, the form asks for a reason why they prefer to take an online exam. To remove any stigma associated with requesting a remote exam, the form explicitly mentions that any reason is valid for taking a remote exam and lists a few examples, including being uncomfortable taking an exam in a large room because of the ongoing pandemic. This was intended to minimize any risk of students taking the exam in-person despite being potentially sick. Friction was further reduced by including a specific humorous reason, stating that the instructor takes exams better with his cat (who made regular appearances during remote lectures) on his lap.

Regardless of exam format, the form also asks if students need accommodations on the exam because of health or disability. In particular, the form checked if students needed extended time on the exam or a low-distraction environment.

The last section of the form asks about alternate times. If the student indicated that they cannot take the exam at the normal time, the standard alternate time (immediately after the main exam) was presented first. If the student also could not make that alternate time, the form presented a few other times, asking students to indicate all the times that worked for them, with a disclaimer that not all times may be used for an alternate exam. The goal of only showing alternate exam times when needed was to discourage students from picking a preferred time and encourage most students to use either the main exam time or the standard alternate time if possible, since much later alternate times can present exam leakage concerns.

Once the form is submitted, a receipt is sent to the student's email. Once a staff member approves and processes the request, a confirmation email is sent to the student's email listing out their exam time, location, and any other relevant information.

The policies of the exam itself were consistent with policies discussed earlier in this report. In-person exams were given on paper, and students were allowed to ask staff for clarifications that would be answered as announcements to the entire class. Remote exams were given through the CS 61A Exam Tool, and answers and clarifications were both submitted through the browser. A complete copy of the exam policies page can be found in the appendix.

7.4 Staff Workflow: Processing Requests

Requests to change exam format were received in a spreadsheet and processed in the order received. Student preferences submitted through the form were processed into a consistent roster format listing each student's exam format (in-person or remote), location, start time, and end time. Because all reasons for remote requests were considered valid, there was no need to manually review documentation or justification for taking a remote exam, which made processing requests more efficient. Once double-checked by a staff member, a mail-merge tool was used to automatically read data fields from the roster and send out a confirmation email to students. Processing requests could be done in large batches to save time.

As seen in Figure 7.3, most submissions to the exam alteration form were students asking to take the exam online. Of the students requesting to take the exam at an alternate time, most were assigned to a remote exam, though this may not fully reflect student preferences because room booking limitations and staff availability limitations forced several alternate time slots to only be proctored remotely. For CS 161, where rooms were not assigned, a small number of students used the form to submit requests to switch in-person exam rooms in order to choose a room that was closer to them or a room that had left-handed desks.

Students who chose to take the exam online were asked to provide a reason, with the guarantee that any reason would be accepted for a remote exam. The example reasons given were "I need to take the exam at an alternate time," "I'm uncomfortable because of COVID," "I have to take care of my family at home," or "remote exams are less stressful for me." Because students were told that all reasons are acceptable, students listed a wide range of reasons for taking a remote exam.

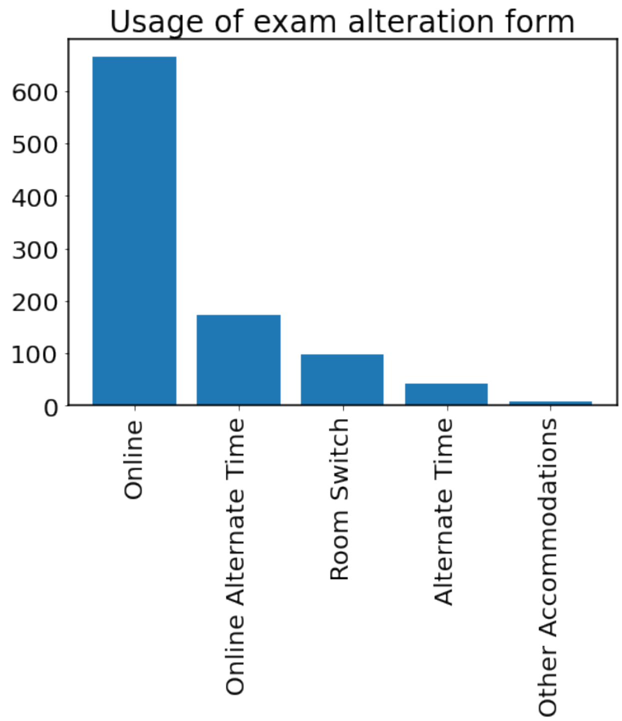


Figure 7.3: Breakdown of the exam format students requested through the exam alteration form.

The listed reasons can be sorted into three main categories, shown in Figure 7.4. Health-related reasons were the most frequently-mentioned reason for taking a remote exam. The next most frequently-mentioned reason was students not physically being near campus. Some students who reported distance as a reason were in the Bay Area but had a relatively long commute to campus, while others were in other states or countries and were studying remotely for the semester. The last main category is students who preferred some aspect of remote exams over in-person exams.

Figure 7.5 has a breakdown of common reasons that students cited when requesting a remote exam. Among students who listed a health-related reason, 24 students specifically mentioned having a family member who was at risk of COVID that they either lived with or would be visiting soon. 18 students mentioned travel plans after the exam, noting that many countries required a negative COVID test for entry, and they did not want to risk getting sick before traveling.

55 students reported being sick or feeling unwell leading up to the exam, and 10 students reported being recently exposed to someone sick, opting for a remote exam to protect themselves and others. Although this is most relevant during the pandemic, a remote option can have health benefits in a post-pandemic world so that sick students do not show up

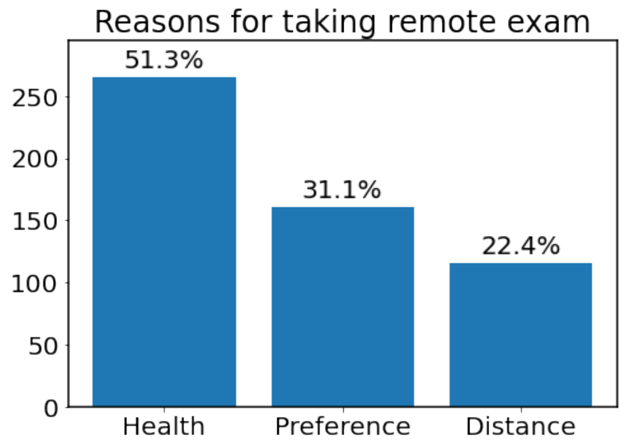


Figure 7.4: Breakdown of reasons listed in requests to take the exam remotely. Note that they sum to over 100% because some people listed multiple reasons.

in a crowded exam room. In addition to the risk of getting others sick, students who are feeling unwell might have a persistent cough or other symptoms that are distracting to other students during the exam. Some of these students noted their distracting symptoms in their remote request as well.

In the category of students who expressed a preference for remote exams, the most commonly listed reason was that remote exams are less stressful (76 students), but this may be because that was one of the examples provided in the form. More specific reasons why students preferred remote exams include the ability to type answers instead of handwriting them, not being around a lot of other students (10 students), larger desk space (12 students), and easier access to bathrooms. 3 students said that they had not taken in-person exams before, so they felt more comfortable with remote exams.

Convenience was a factor behind remote exams for some students. Because it was raining heavily the day of one exam, 15 students opted for a remote exam to avoid going outside in inclement weather. 5 students preferred remote exams because of timing conflicts that made remote exams more convenient for them. For example, some students had interviews or other exams that were happening elsewhere directly before or after the exam, so they preferred to take the exam remotely to eliminate commuting time. The pandemic also caused 9 students to note that wearing a mask for an extended amount of time was uncomfortable, and they would prefer taking the exam at home without a mask on.

CS 61C's exam was scheduled to take place from 7:00 PM to 10:00 PM, so 15 students requested a remote exam because they felt unsafe going home from campus that late at night. Although this issue with late-night exams had been reported before during fully in-person semesters, remote exams are a promising long-term solution to making sure that students feel safe during their exam.

Reason	**CS 161**	**CS 61C**
Less stressful	27	49
Sick or not feeling well	25	30
More comfortable	14	14
Avoid getting family sick	9	15
Need to travel soon	9	9
Bad weather	8	7
Late at night	0	15
Less distracting	6	6
Larger desk	0	12
Exposed to COVID	2	8
Work worse in large crowds	4	6
Masks uncomfortable	5	4
Time conflict	2	3
Remote exams are more familiar	0	3

Figure 7.5: Number of students who listed each common reason for taking remote exams from students in CS 161 (8:00 AM exam) and CS 61C (7:00 PM exam).

Of the students who preferred a remote exam, some mentioned specific reasons why remote exams were more accessible for them. For example, 12 students mentioned that they were less distracted in a remote exam, including a few students with ADHD who said they could focus better in a room by themselves. Students with medical conditions that require frequent bathroom trips reported that having a bathroom nearby at home affected their decision to opt for a remote exam. Although not everyone prefers remote exams, these cases show how remote exams can be used to improve accessibility and allow disadvantaged students to take the exam in an environment where they are most likely to succeed.

7.5 Staff Workflow: Exam Writing

Exams were first drafted in a human-readable format, such as Google Docs, then converted to a LaTeX PDF for in-person students and imported into Exam Tool for remote students. As of the Fall 2021 semester, there was no way to automatically convert between LaTeX and the Exam Tool format (Markdown with additional directives for randomization), so the conversion had to be done by hand, which led to a few mistakes. One alternative to this approach that would minimize human errors would be to print out the Markdown directly through Exam Tool into a PDF, but this provided less control over page layout than manual LaTeX, and page layout was sometimes important in ensuring the same exam environment for all students (as discussed below). Future long-term exam delivery tools such as PrairieLearn may want to streamline the process of formatting an exam such that students taking the exam on-paper or in-browser have the same experience.

When writing the exam, fairness between in-person and remote test takers had to be considered, as discussed earlier in this section. A few examples of these considerations are listed below.

CS 61C often has exam questions where students need to fill in blanks in a given code snippet. Paper exams usually ask students to write directly into the blanks, but this is more difficult in the Exam Tool format, where only short-answer blanks were available, and fill-in-the-blank coding questions were not yet supported. Fully-remote exams often circumvented this by providing one short-answer box per blank and asking students to write their answers in the short-answer boxes, separate from the actual code. However, being able to write directly in the code blanks might provide an advantage to writing answers in text boxes after the code. To balance this for remote students, the question code (with blanks) was given in a large text box, and students could type their answers, replacing the blanks. The question code was reproduced in the exam in case students overwrote some provided code.

CS 161 exams usually provide a C appendix containing function definitions of any C functions that might be unfamiliar to students. When exams were fully in-person, this appendix was usually attached at the end of the exam for all students. Some semesters allowed students to detach the appendix, while other semesters required the appendix to stay within the packet. In either case, all students referenced the appendix in the same place. However, with remote exams, students taking the exam online could potentially duplicate their browser tab to view the appendix side-by-side with the exam question. To ensure that in-person students were not disadvantaged, the appendix was printed separately from the exam packet and distributed to students so that in-person students could also reference the appendix and the exam question side-by-side.

A similar issue involved questions being split across multiple pages, especially when the entire question relied on a code snippet or a diagram provided at the beginning of the question. Students taking the exam remotely could split their browser to view the question setup throughout the entire question, or they could print out the exam to keep the question setup in view throughout the exam. To balance this for in-person students, the question setup was repeated on subsequent pages so that students could view the question setup and later parts of the question on the same page without flipping through the exam.

It is unclear whether these more minor quality-of-life improvements led to significant differences in exam scores; future studies could investigate how logistical differences between in-person and remote exams affect student exam performance and the student exam experience.

7.6 Staff Workflow: Clarifications

A small team of TAs and instructors who were most familiar with the exam met in a room (or remotely through videoconferencing), separate from students, to discuss clarification requests and publish any clarifications to the entire class. This setup is more similar to the clarification process during fully-remote exams than the process during fully in-person exams. Since more clarification requests were expected in a hybrid exam than a fully in-person exam (because of the students taking the exam remotely), and hybrid exams had two different clarification queues to be processed (in-person and remote), having a separate room allowed for more open discussion without disturbing students and more organization in processing the requests in a timely manner. Also, having a centralized clarifications meeting ensured more consistency in answering clarifications from different exam rooms and online exams, compared to the more informal setup in fully in-person exams where most clarification requests were answered by different staff members.

Proctors in exam rooms received questions from students taking the exam in-person, transcribed the question, and submitted the question to the clarifications team. Submitted questions were sent through Slack. It would also be possible to submit questions through Exam Tool, which would merge all clarification questions into a single queue, but Slack was selected because some proctors used their phones to submit clarification requests, and Exam Tool is more difficult to use on a phone.

The clarifications team processed both queues in first-in, first-out order. If the volume of clarification requests was too high and the team was large enough, the team could split up and work on both queues separately. If a clarification announcement had to be issued to the entire class, it was submitted as an announcement through Exam Tool for remote students, and posted on a Google Docs page being projected in all the exam rooms. Another approach would be to project the Exam Tool announcements in exam rooms. However, since some clarifications might only be relevant to one of the two exam formats (e.g. a typo is caught only on the remote version of the exam), two different sets of announcements were maintained to avoid confusion. Any content-related clarification was announced to all students, and logistics clarifications that only affected in-person or remote students were only announced to the relevant students.

As expected, there were more clarifications submitted by remote students, as shown in Figure 7.6. When adjusted for the fact that there were fewer students taking the exam remotely, the difference is even more pronounced, as shown in Figure 7.7. It should be noted that these numbers include logistics questions for remote students, but not for in-person students, since in-person proctors could answer logistics questions without officially submitting a clarification request.

7.7 Exam Grading and Cheat Detection

For grading, in-person exams were scanned and uploaded to Gradescope, and remote exams used Exam Tool to generate PDFs of answers that were then uploaded to Gradescope. Because the PDFs produced by Exam Tool were formatted differently from the paper exams, Gradescope's automatic grading tools needed to be configured twice for every question on the exam, which added to the staff overhead when grading exams. Also, Gradescope is

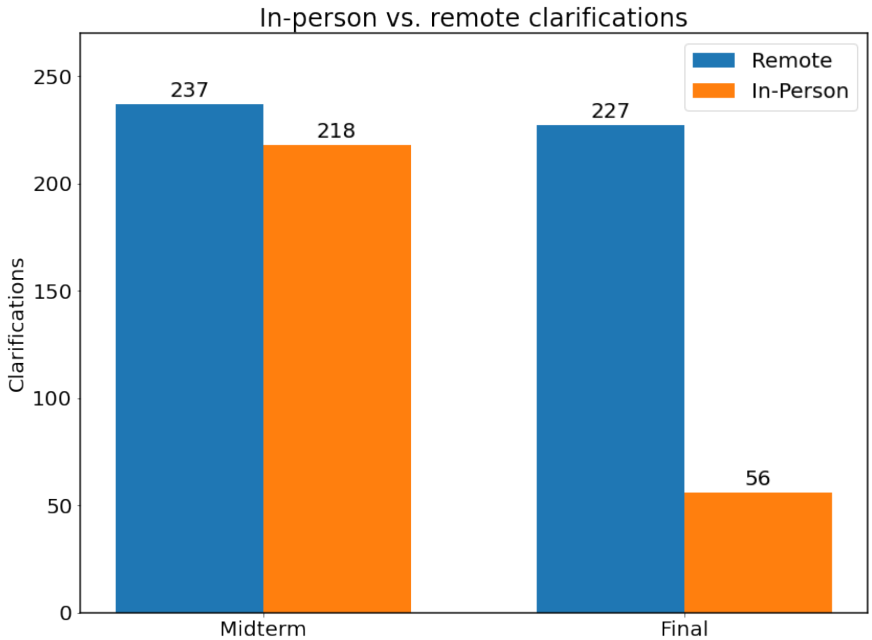


Figure 7.6: Number of clarification requests submitted by remote students and in-person students.

configured for handwritten exams, so despite the fact that remote exam submissions were text that could be processed and possibly graded automatically, it was difficult to leverage this on Gradescope. Future remote exam platforms could support exam grading that works for both in-person and remote exams simultaneously, while allowing typed answers to be processed for easier grading.

Hybrid exams provide a unique opportunity to assess cheating rates between in-person and remote exams; this is discussed further in the cheating detection section later in this report.

7.8 Exam Grade Analysis

A surprising result from all four exams conducted is that even with a large sample size, remote students consistently averaged several points lower on the exam than in-person students (Figures 7.8 and 7.9). This seems to contradict the hypothesis that it is easier to cheat on remote exams, which would in theory result in a higher average on remote exams.

One possible explanation for this phenomenon is that students are not being randomly sorted into in-person and remote exams, and are instead self-selecting which exam format they prefer. Students who opt for a remote exam might have other external circumstances

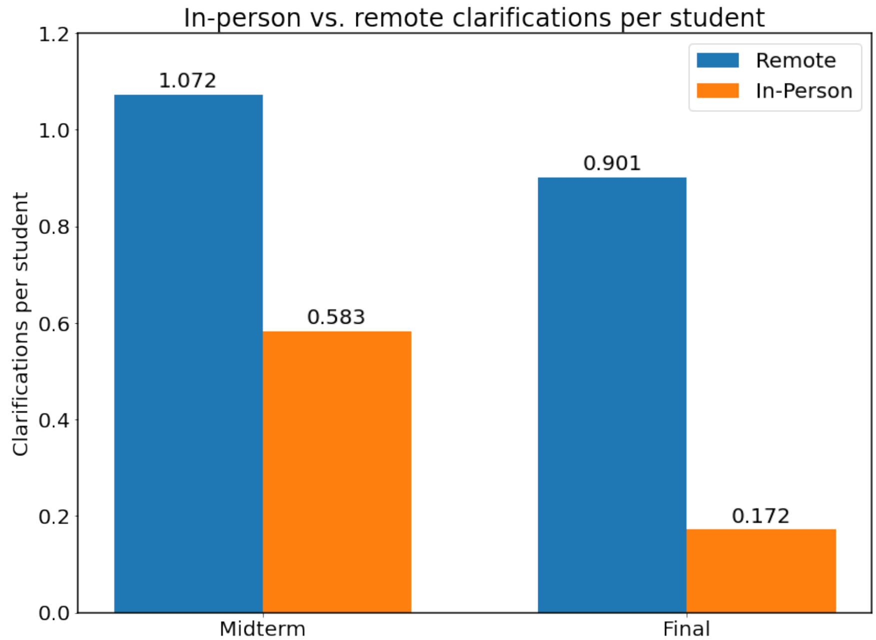


Figure 7.7: Number of clarification requests submitted per student for remote students and in-person students.

that are negatively affecting their exam score, such as getting sick, being in quarantine from COVID-19 exposure, or living away from campus and being isolated from their peers.

At the same time, this result may suggest that under this particular hybrid exam model, the advantages of taking the exam remotely, such as access to a digital exam and the ability to take the exam at home or in a location of their choice, might not be statistically significant enough to cause a difference in the exam averages of a large class. Further work may be able to investigate how the benefits and drawbacks of remote exams and in-person exams differ between individual students.

7.9 Summary and Future Work

Hybrid exams went relatively smoothly in this first attempt, in part thanks to the infrastructure for remote exams that had already been developed in previous semesters. However, many backend tools had unnecessary manual overhead that would prevent them from scaling to larger classes efficiently. Future work on developing remote exam platforms should keep these logistics challenges in mind and streamline them so that hybrid exams can be held without the overhead of managing two different exam setups with different software and infrastructure.

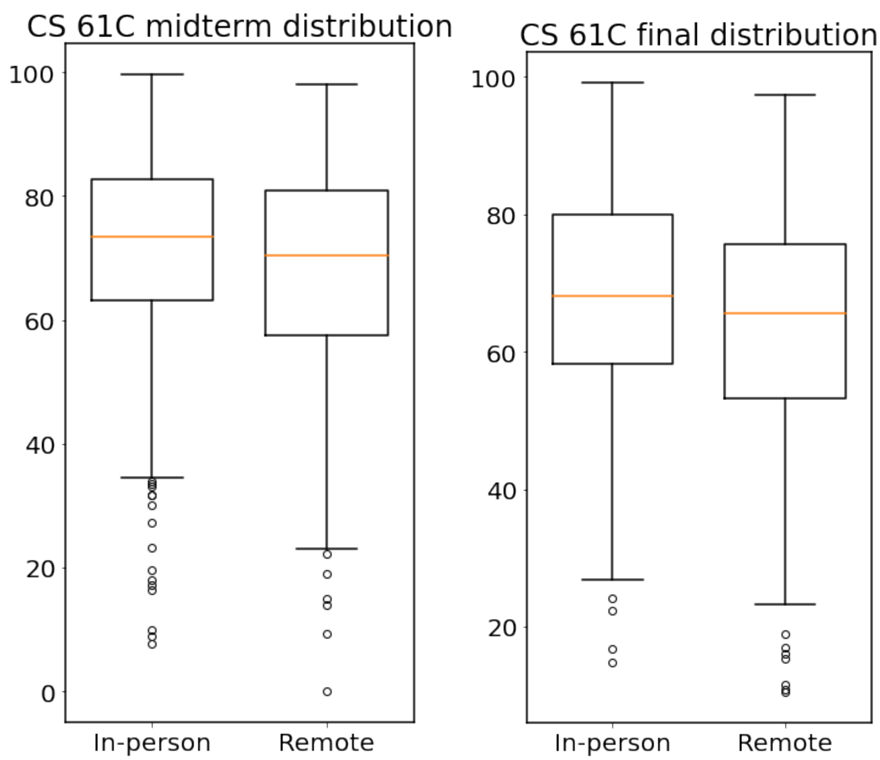


Figure 7.8: Exam score distributions for the CS 61C midterm and final, split by exam format (in-person or remote). Note that the distribution for remote students is consistently lower.

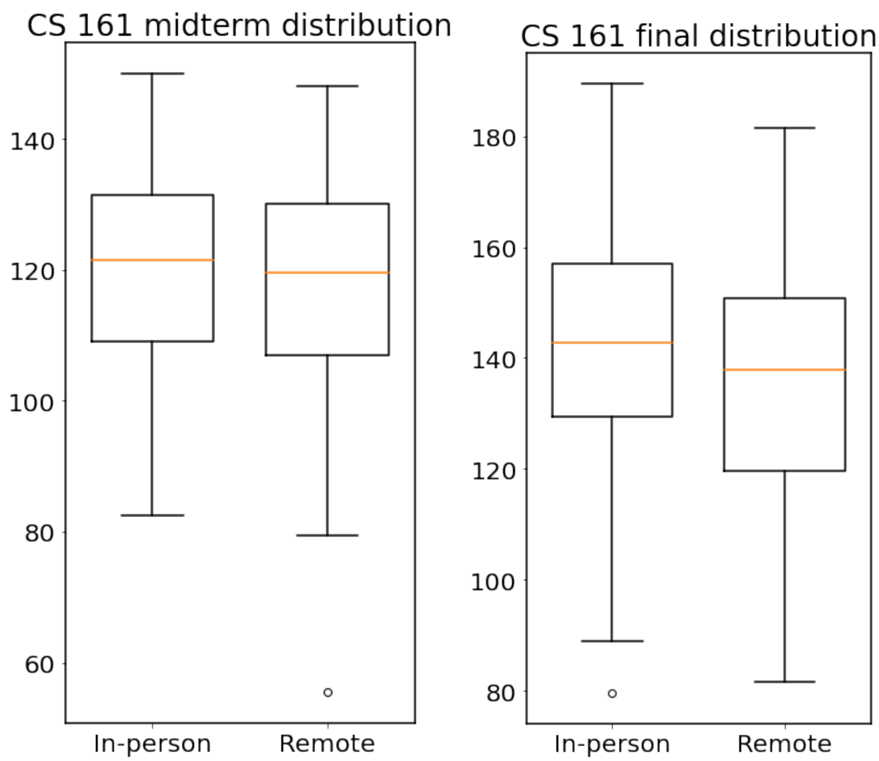


Figure 7.9: Exam score distributions for the CS 161 midterm and final, split by exam format (in-person or remote). Note that the distribution for remote students is consistently lower.

Chapter 8

Accessibility and Equity

The progress made in the space of remote exams has promising implications for the future of making computer science classes more accessible and equitable for all students. The tools and policies developed in the past two years since the start of the pandemic have already shown immediate benefits for students from historically underprivileged groups. However, without careful design that explicitly considers the needs of disadvantaged or disabled students, newly-developed remote exam platforms and proctoring policies may exacerbate existing inequities instead of solving them. Although in-person exams will likely remain the primary testing format after the pandemic, retaining the lessons learned from remote exams and using remote exams in conjunction with in-person exams can make exams more accessible and equitable for all students.

8.1 Inequities of Remote Exams

Remote learning during the pandemic has exacerbated many existing disparities in higher education [24]. In particular, students with disabilities, who already faced disproportionately many barriers to access before the pandemic, were often inadequately supported during the pandemic [9]. Although the sudden switch to remote learning caused many vulnerable student populations to be left behind, remote learning itself is not inherently inaccessible, and post-pandemic efforts can leverage the benefits of remote learning to specifically serve disadvantaged groups [41]. For example, although some students felt that remote learning through Zoom provided an inferior learning experience, others highlighted the flexibility of remote learning as a benefit [43]. This additional flexibility can be beneficial for making classes more accessible to students who may not be able to devote all their time to school.

Remote exams follow a similar pattern as remote learning as a whole: the sudden transition to remote exams negatively affected students from marginalized groups, and remote exams may not be the best option for all students in the long term. However, in a post-pandemic world, remote exams provide unique benefits that can make it a useful tool for serving disadvantaged students. When designing remote exams as a long-term tool to improve educational equity, it is important to consider both the ways in which remote exams increased educational inequality during the pandemic, and how remote exams can be used to specifically improve the learning experience for disadvantaged student groups.

In particular, the gap in access to technology resources, which disproportionately impacts students from lower socioeconomic backgrounds, makes it more difficult to access online learning resources, including remote exams. Before the start of each semester, students from CS 161 were surveyed about their access to some resources that may be needed to take remote exams. The results of one of these surveys is detailed in Figure 8.1.

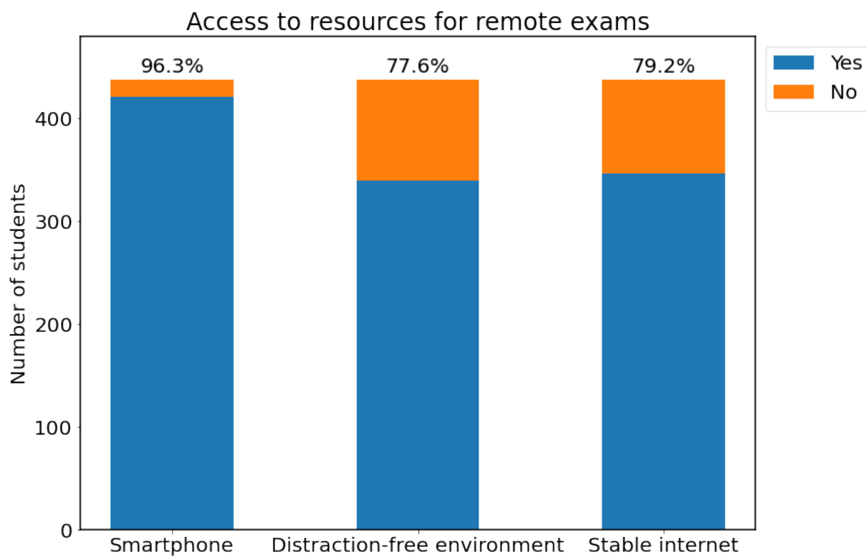


Figure 8.1: Fall 2020 CS 161 (upper-division computer security, 450 students) survey showing students' self-reported access to resources needed or helpful for an online exam.

Although a majority of students indicated that they had sufficient equipment for remote proctoring, a few students reported not having basic equipment, such as a smartphone with a camera. The proctoring logistics offered enough alternatives, such as a webcam proctoring setup, so that all students were able to take the exam with some video equipment that they had access to. In the event that someone did not have any video equipment they could use, the logistics asked students to reach out to instructors privately, though this did not happen. For schools where students may have less access to video equipment, such as places where smartphone ownership and Internet access are not as widespread, these technological considerations would need to be taken into account when implementing remote proctored exams.

Although a stable Internet connection and a quiet home environment are not strictly required for a remote exam, a significant number of students indicated that they lacked one or both of these. This raises fairness concerns, as a student with a better environment to take the exam may have an unfair advantage on the exam. During the pandemic, it was difficult to find a robust solution to this problem, as most people were in lockdown and most public spaces were closed, though some students did opt to take the exam outside or in a library (if open). Students who reported significant trouble or distractions in their exam environment may have been given additional time on the exam as compensation, though this happened on a case-by-case basis. In a post-pandemic environment, ensuring that all students can find a quiet place with stable Internet access for taking the exam will be important to ensure exam equity. This could include reserving classrooms on campus for students to take remote exams, a dedicated testing center for computer exams (as discussed in the hybrid exams section), or helping students get set up in nearby locations such as libraries.

8.2 Remote Proctoring Accommodations

Examining new accommodations created specifically to help disadvantaged students navigate remote exams can help identify additional work needed to make remote exams more equitable. Mandatory remote proctoring is a relatively new development in EECS, so accessibility concerns with proctoring policies have not been studied in detail yet. Nevertheless, in response to student feedback about remote proctoring policies, the Disabled Students' Program at Berkeley has introduced a new "no COVID-19 video proctoring" accommodation for students who would be unfairly disadvantaged if required to follow remote proctoring procedures.

8.3 Scalable Exam Accommodations

Exam accommodations for disabled students and students with other extenuating circumstances can be time-consuming for staff to set up, since each case must be reviewed individually and accommodations can be different for every student. This makes it difficult to scale accommodations to large classes, and larger classes are often forced to give less specialized accommodations that do not fully meet a student's individual needs. Remote exams offer a scalable solution to providing more individualized accommodations, because they give students more freedom to set up an exam environment that works for them, while offloading logistics work from staff.

Reduced Distraction Exams

UC Berkeley's DSP program has a "reduced distraction" accommodation for exams, which lets the student take the exam in a quiet, distraction-free environment. The definition of a reduced-distraction environment can vary between students, but most traditional testing environments do not qualify as reduced-testing environments. For example, a smaller testing room with DSP students may have students and staff frequently entering and leaving the room. Students might also be asking clarification questions to staff, and staff might be making announcements. There may also be background noise from adjacent classrooms.

Providing reduced-distraction environments to many students in an in-person exam is difficult, because it would require many different rooms and additional proctors per room. However, remote exams can be used to create more scalable reduced-distraction exams. Students can take the exam from home or any reduced distraction location on campus and be proctored remotely. This would also allow students to take the exam in a room by themselves, without a proctor.

8.4 Case Study: In-Person Webcam Proctoring

Motivation and Design Philosophy

In the fall 2021 semester, there was frequently a need for more physical exam rooms than proctors were available. Because of the ongoing pandemic, many TAs worked fully remotely and were unable to proctor exams in-person, leading to a general shortage of in-person exam proctors. Also, TAs may test positive or be exposed to COVID-19, requiring a quarantine period and making them unavailable for in-person exam proctoring.

Likewise, the pandemic affected students and required the use of additional exam rooms. Students who were unvaccinated, recently tested positive, or were exposed to COVID-19, but were unable to easily take a remote exam (e.g. Internet issues, distracting home environment) preferred to take the exam in-person, but would need to be placed in a separate exam room by themselves for the safety of staff and other students. Students who planned to take the exam in-person regularly, but showed symptoms during the exam (e.g. coughing) might also need to be relocated to an exam room by themselves.

Beyond the pandemic, students with reduced-distraction DSP accommodations, as described earlier, may need to be in a room by themselves or with a small number of other students in order to fully satisfy their accommodation. Sometimes, the presence of a proctor in the room may be potentially distracting to the student as well.

Proctor shortages will persist after the pandemic as well, especially for exams taken at alternate times and exams with additional time. For a 3-hour final exam, a student with a 200% time accommodation taking the exam at an alternate time needs 6 hours of proctoring, which scales poorly if there are multiple different alternate times that result in ongoing exams through an entire day.

Historically, these problems were usually solved by putting all alternate-time students (with or without DSP accommodations) in the same exam room and restricting the amount of alternate times available to students. Although this helps to ensure that proctors are available, these solutions come at the expense of student accessibility and disproportionately impact disadvantaged students.

To support more scalable in-person proctoring across different exam rooms, staff from several EECS courses developed a proctoring setup where webcams installed in each exam room allow a single proctor to manage exams in several different exam rooms, while minimizing the amount of face-to-face contact.

Student Workflow

When students arrive at their designated exam rooms, the room will have no proctor, but a projection describing the webcam proctoring logistics. The proctor(s) will usually be standing outside the exam rooms to help any confused students. The student is free to prepare for the exam in their room at this time.

Near the exam start time, a proctor will enter the exam room to hand out exams, answer any last-minute logistics questions, and start the exam timer.

During the exam, a laptop with a webcam is placed at the front of the room to monitor students. This laptop also has a chat connection (e.g. through Zoom) to the proctor. If a student has any questions or concerns, they can "raise their hand" by walking up to the laptop and typing their question into the laptop. This can be used for content-related clarification questions and logistics questions such as requesting to use the bathroom or turning in the exam early. If needed, the proctor can walk over to the exam room to resolve the issue as they would for a standard in-person exam.

At the end of the exam, the proctor re-enters the room to call time and collect the student's exam. Once the exam is turned in, the student is free to leave.

Staff Workflow

The main overhead for webcam proctoring is the setup that has to be done beforehand, though this setup usually takes no longer than one hour and can be parallelized with multiple TAs.

When planning for the exam, multiple rooms need to be reserved for webcam proctoring. These can usually be smaller classrooms or conference rooms, as each room will only contain a few students. Rooms should be nearby (e.g. adjacent classrooms on the same floor of a building) so that the proctor can reach all the rooms in a minute or two if needed. It helps to reserve one extra room for the proctor if available. When reserving room times, it helps to reserve the rooms one hour earlier than the exam start time so that the rooms can be set up ahead of time.

In order to meet reduced distraction accommodations, we decided to book smaller classrooms (capacity of 20-30 students) on the basement floor. This ensured that fewer students would be walking in the hallways and distracting students (compared to the busier ground floor). Also, the classrooms on the basement floor provide better sound insulation to reduce ambient noise that may be distracting to some students. A room can have as few as one student inside, and each room usually has no more than 8-9 students in order to meet reduced distraction accommodations.

One laptop is needed for each exam room. EECS staff loaned Chromebooks from the university library for this purpose, though personal laptops from staff could be used if available. For reduced-distraction exams, “do not disturb” signs were printed ahead of time as well.

An hour before the exam, the proctor(s) go to each room to set up the webcam proctoring. The laptop joins a Zoom call, which facilitates both the webcam proctoring from the TA and the text channel for students to contact the proctor. The laptop is also used to project the timer and clarifications for students, as in a standard in-person exam. Before the exam starts, the clarifications page is usually replaced with a page informing students of the webcam proctoring setup in order to avoid confusion. At this point, the rooms are ready for students. Note that the actual exams have not been given to students yet.

The proctor goes to each room to distribute and start the exam timer. This does mean that each room starts at slightly different times, but as long as the exam rooms are all nearby (e.g. on the same floor of the same building), it is usually possible to start all the exams within 10 minutes of each other. Having multiple proctors at the very beginning of the exam can help speed up this process, and it only requires the additional proctor to be present for about 10 minutes.

During the exam, the proctor monitors the exam rooms through the Zoom call, similar to how synchronous remote exams are proctored. Student concerns are addressed through Zoom chat, and the proctor can resolve them depending on the policies of the specific exam. Because of the pandemic, most concerns were resolved without face-to-face contact if possible: for example, if a student needs to use the bathroom, they can leave their phone in sight of the webcam and leave, without ever coming in contact with the proctor or other students.

At the end of the exam, the proctor revisits each room to collect exams. It is usually possible for the proctor to visit each room as the exam is ending, if done in the same order that exams were distributed, though in the worst case, students may need to wait a few minutes before their exam is collected. The webcam proctoring can be used to ensure that the student doesn’t work after their exam is over.

Analysis and Future Work

Students who have used this proctoring setup have provided positive feedback on its ability to meet their accommodations without the additional student-side overhead associated with remote proctoring. Also, the exam security seems to be comparable to, if not better than a standard in-person exam. (The additional security might come from the fact that students in their own exam room have no other students to collude with.) None of the cheat detection case studies described later in this report identified any students who used webcam proctoring as having cheated. However, the small sample size of students means that further work is needed to analyze its security.

Future work can look into adapting this policy for post-pandemic times, when avoiding face-to-face interaction is less of a concern, but maintaining low distraction environments continues to be a priority for DSP students.

Chapter 9

Cheating Prevention and Detection

9.1 Online Cheating

Even with strict proctoring, remote exams have more cheating vulnerabilities than in-person exams. When designing cheating detection strategies, it is important to first consider the types of cheating we are trying to detect.

Most notably, in an online exam, students can interact with anything on the Internet. This includes existing class resources (e.g. lecture slides), resources outside of the class, search engines [5], Q&A forums (e.g. Stack Overflow, Chegg) [29], and their friends (e.g. chats, collaborative text editors like Google Docs). This presents a new set of cheating methods that may not be detectable using traditional in-person exam cheating detection methods.

However, online exams also provide instructors with more data that can be analyzed for evidence of cheating. Many new cheat detection methods take advantage of this additional data that would not be available on an in-person exam.

As a result of the additional cheating methods and the additional tools for detecting cheating, recorded cases of academic misconduct increased significantly during the pandemic (Figure 9.1). According to the university's office of student conduct, cheating cases increased the most in classes where a student's grade is a prerequisite to declaring a major. This trend was seen in all departments, including EECS classes.

9.2 Prevention: Exam Randomization

With remote proctoring, a student trying to cheat has limited communication bandwidth with others. For example, a student trying to cheat during a bathroom break only has a few minutes to cheat before their extended break would raise suspicion. A student talking to someone else over a phone call might only be able to say a few short words and cannot openly discuss the entire exam. A natural strategy to make this type of cheating harder is to increase the amount of information that must be communicated in order to share answers.

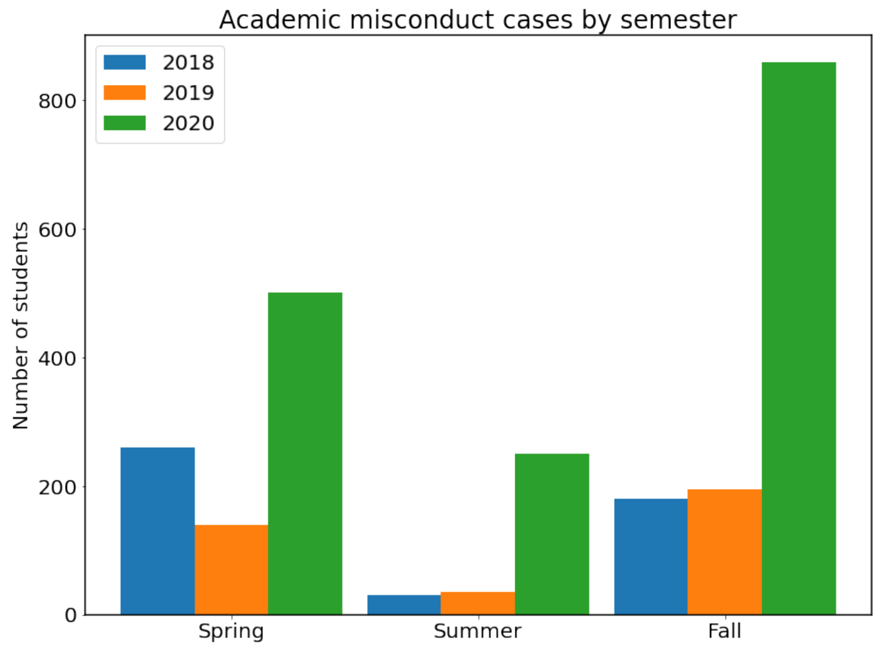


Figure 9.1: Number of academic misconduct cases across the entire university per semester. Note the significant increases in 2020 when classes moved online.

Past studies have shown that randomizing the order of the exam makes sharing answers more difficult [7]. Without randomization, it may be possible to identify an answer choice on a specific subpart of a specific question with a number and letter, like (A) on 9.3. However, if the questions, subparts, and answer choices are randomized, this will likely correspond to a completely different question and answer choice on each student's exam. Instead, students will need to communicate longer descriptive information such as the name of the question, some phrasing to identify the relevant subpart, and the entire text of the answer choice. While still possible, this is not only more difficult to share without getting caught, but is also more time-consuming, which makes it a less appealing option. The degree of exam randomization must be balanced with keeping the logistics of derandomization manageable, maintaining the coherence and structure of the exam, and preserving exam fairness.

Earlier in the pandemic, some remote exams used platforms like Gradescope that do not natively support randomization. As a result, derandomization required external scripts and possibly some processing by hand. In these cases, aggressive randomization may have required too many staff-hours or too many logistical problems for staff and students to be fully implemented. However, with the development of Exam Tool and PrairieLearn, which support customizable randomization and derandomization, highly randomized exams are easier to generate and grade.

The degree of exam randomization may be limited depending on the exam structure.

Some exams have questions arranged from easiest to hardest, which is at odds with question randomization. Additionally, some questions might be arranged in sequential order, where a student must answer one subpart before answering the next subpart. These questions cannot have subpart randomization.

In some cases, randomization is possible, but the additional entropy may not be worth confusing students. For example, multiple-choice questions might have several subparts with the same set of answer choices. Giving a student the answer choices in a different order for every subpart might obfuscate the fact that every subpart has the same set of answer choices. This randomization may also result in slightly unfair exams, because some students may get the answer choices in the same order on different subparts, while other students get the answer choices in a different order.

Even if the exam is not intentionally arranged from the easiest question to the hardest question, the randomization of questions may result in some unfair exams. Past studies have demonstrated varying results on whether students who see questions in sequential order have an advantage over students who see questions in random order [4, 35]. The use of remote exam platforms may further complicate this problem, so additional studies may be needed. For example, assuming most students work through the exam in order, a student who receives the hardest question first may be at a disadvantage compared to a student who receives the hardest question last. This imbalance may be reduced by telling students ahead of time that they may receive questions in a different order, encouraging them to work out of order, or reducing the randomization entropy so that every student receives the hardest question at the end of the exam.

9.3 Detection: Versioning

Note: Some parts of this section have been redacted for publication.

Random parameters may need to be carefully chosen, especially if they involve calculations or math. Different numerical parameters should not make a question harder or easier for students. For example, in a question involving adding 4 to a hexadecimal number, "0x5C" would be more difficult than "0x40," because the student with "0x5C" would need to understand how to carry the one in addition, while the student with "0x40" only needs to add 0+4. Similarly, in a question involving dividing by 3, "819" would be a more difficult keyword than "900." In questions with more complex calculations, it may be important to fully solve the question with every different keyword number in order to ensure that the calculations aren't more difficult on certain versions. Some EECS exams that used numerical cheat traps had imbalanced exam versions depending on the combination of numbers.

Creating different exam versions with cheat trap questions, derandomizing exam versions, and running detection with cheat traps requires some additional logistical work. Gradescope and PrairieLearn do not natively support cheat trap questions yet, but Exam Tool does have support for exam versions. Canvas does have question banks, which can be used to implement different versions of questions, but it does not have automatic cheat trap detection.

9.4 Detection: IDs and Watermarking

Assigning a unique ID to every student follows a similar idea as versioning. If a student copies an answer from another student or tries to post their question online to solicit answers, the ID will provide enough information to identify the student from whom the question or answer was sourced from. IDs are particularly useful for detecting when students take a picture or screenshot of the exam to solicit answers to the questions.

IDs can be placed in the exam in different ways. One simple implementation involves writing the ID in small print in whitespace throughout the exam, such as between questions. The IDs should be placed carefully, or a note should be placed around the ID to avoid students accidentally interpreting the ID as relevant to the exam content. However, placing IDs between questions might be circumvented if a student takes a picture of a question and crops the ID out of the picture or screenshot. On the other hand, placing IDs in the middle of a question might be unnecessarily confusing for honest students, so a trade-off between usability and security exists when finding a location for IDs.

Another way to embed IDs into questions is to incorporate a unique ID into each student's versioned exam. Embedding unique IDs in questions can be difficult for larger classes, when IDs need more entropy to be unique for every student. While it may be possible to create a few different variants of a question (e.g. changing a name in a word problem between Alice, Bob, and Charlie), it may be more difficult to produce hundreds or thousands of variants without introducing potential concerns about fairness or false positives. For example, a variable name in a coding question could have a unique ID number appended to it (e.g. tree6253), but a student could mistype this number and cause a false positive. Also, the additional overhead of using a variable name with a meaningless number appended may not be worth the cheat detection capabilities.

Watermarking the exam involves placing a unique pattern in the background of the exam, so that if a student takes a picture of their exam, the watermark can be used to uniquely identify whose exam the picture originated from. The CS 61A Exam Tool implemented watermarking and detection by placing a unique pattern of dots on the exam page (Figure 9.2), a technique inspired by color printers.

Again, there is a tradeoff between usability and security with watermarks: if the watermark is too prominent, it may be too distracting for students, and the clutter of dots on screen may lead to accessibility problems. For this reason, some classes opted to use very faint watermarks, and some classes used no watermarks at all, relying on other types of IDs or versioning instead.

9.5 Detection: Comparing Similar Exams

The strategy of comparing similar exams has been used before on in-person exams. Not only is this strategy still effective with online exams, but the fact that online exams used typed

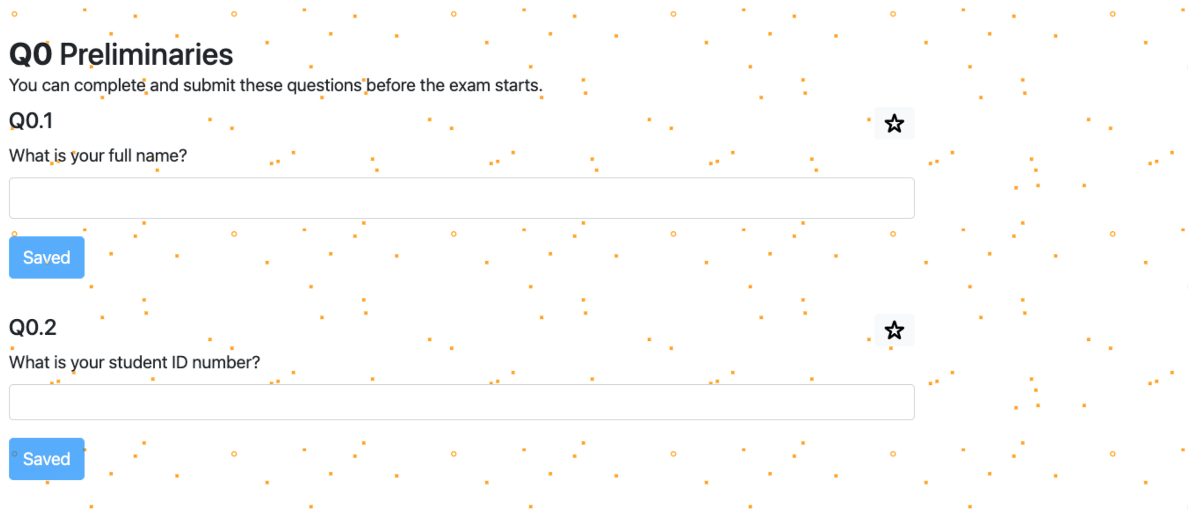


Figure 9.2: A sample exam question on Exam Tool with a watermark in the background.

answers instead of handwritten answers makes analyzing answers easier and more scalable for large classes.

Although many automated methods of correlating exam answers exist, this report will focus on Q-SID (Question-Score Identity Detection), a method of detecting similar exams developed at UCLA in collaboration with Berkeley's EECS department during the pandemic. Q-SID examines the breakdown of scores a student receives on each question of the exam and looks for other students who have a very similar score breakdown. Using question scores instead of answers is a simple way to compare cheating rates between in-person and remote exams: handwritten answers are difficult to process and compare, but the scores given to those answers is a good indicator of how the student answered the question. Furthermore, question score data is available for fully in-person exams from before the pandemic, providing a useful baseline for analyzing cheating rates on remote exams.

Note that Q-SID, and the method of comparing similar exams in general, is most helpful for detecting collusion, where students work together on the exam to come up with the same answers, or one student copies off another student resulting in similar answers. It is less helpful for detecting other forms of cheating, such as when students access disallowed resources (e.g. searching for answers on the Internet), or when students receive a copy of the exam ahead of time, as these are less likely to result in groups of students having similar answers.

Semester	Exam	Format	Number of Submissions
Summer 2019	Midterm 1	In-person	56
Summer 2019	Midterm 2	In-person	53
Summer 2019	Final	In-person	54
Fall 2019	Midterm 1	In-person	475
Fall 2019	Midterm 2	In-person	462
Fall 2019	Final	In-person	463
Spring 2020	Midterm 1	In-person	630
Spring 2020	Midterm 2	Remote unproctored	617
Spring 2020	Final	Remote unproctored	610
Summer 2020	Midterm	Remote proctored	85
Summer 2020	Final	Remote proctored	82
Fall 2020	Midterm	Remote proctored	446
Fall 2020	Final	Remote proctored	408
Spring 2021	Midterm	Remote proctored	436
Spring 2021	Final	Remote proctored	419

Figure 9.3: Comparison of features on various remote exam platforms.

9.6 Q-SID Case Study: CS 161

In this study, Q-SID was run on exam score data from 15 different exams over 6 semesters of CS 161 to find evidence of collusion. This dataset includes in-person exams, unproctored remote exams, and proctored remote exams, so they provide an opportunity to compare collusion rates between the different exam formats. Details of each exam in the dataset are provided in Figure 9.3.

For each student, Q-SID finds another student in the class with the most number of question scores in common, and assigns the student a collusion score based on how similar their exam is to the nearest student in the class. Higher collusion scores correspond to more similar exams and a higher likelihood of cheating. A background false positive rate of 0.3% is expected, which means that Q-SID will incorrectly flag 0.3% of non-colluding students as having a collusion score high enough to indicate cheating. However, the higher the collusion score is, the less likely the case is a false positive, and collusion scores above 2.0 are unlikely to be the result of a false positive.

Q-SID uses a control exam dataset to generate the expected distribution of collusion scores on a strictly proctored exam believed to have no collusion. The distribution of collusion scores for each of the exams can then be compared to this control distribution to get an approximate estimate of cheating rates.

The most important part of these distributions is the tail, where collusion scores are abnormally high. The tails on the in-person exams (Figure 9.4) are comparable to those of

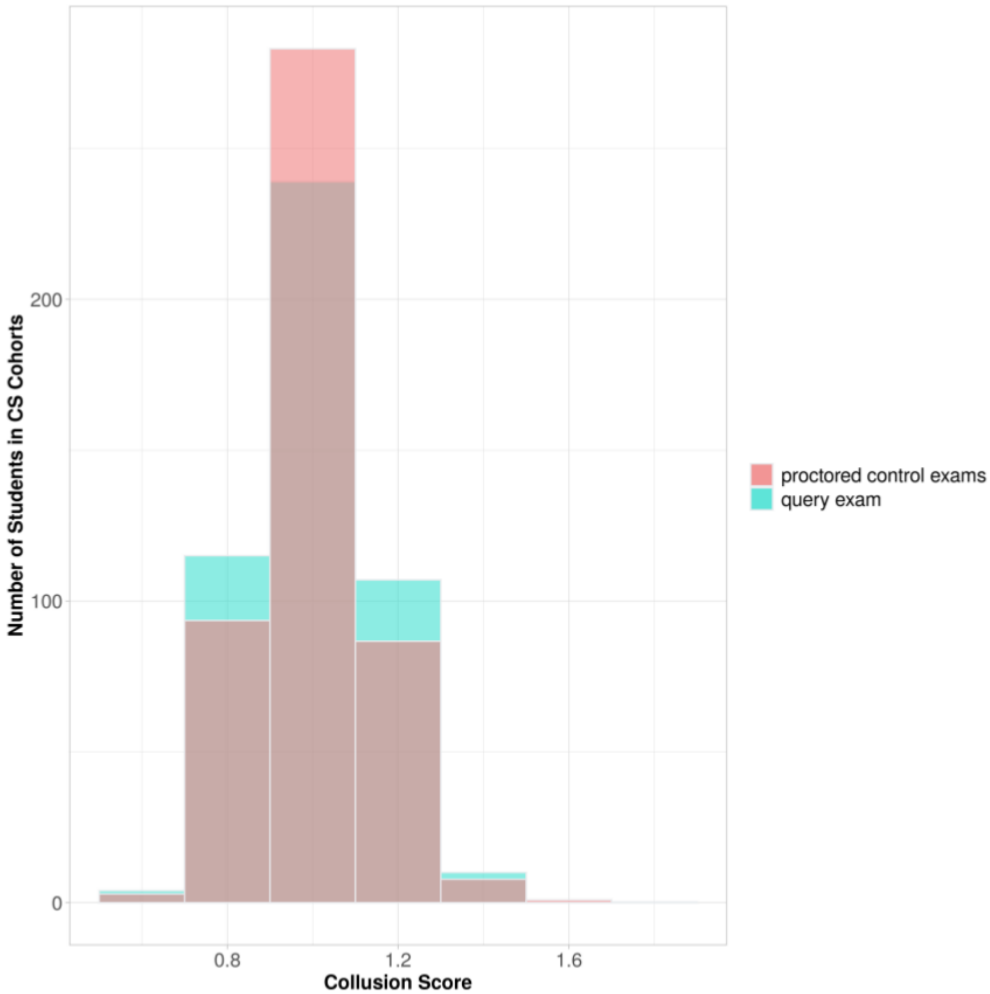


Figure 9.4: Distribution of collusion scores on an in-person exam (CS 161, Fall 2019). Note that the distribution is similar to the distribution of the control exam dataset.

the control distribution, with very few outliers, which suggests that in-person exams had little to no collusion, even when factoring in the potential exam leakage from offering alternate exams.

The unproctored remote exams (Figure 9.5) show the opposite result, with many cases of abnormally high collusion scores, including scores above 2.0, which were not present in the in-person exams distributions.

The proctored remote exams (Figure 9.6) show a decrease in the number of outliers compared to the unproctored remote exams. One notable result here is that the outliers in the proctored remote exams have scores above 2.0, unlike the in-person exams. This suggests that a few students were able to somehow circumvent the proctoring policy to

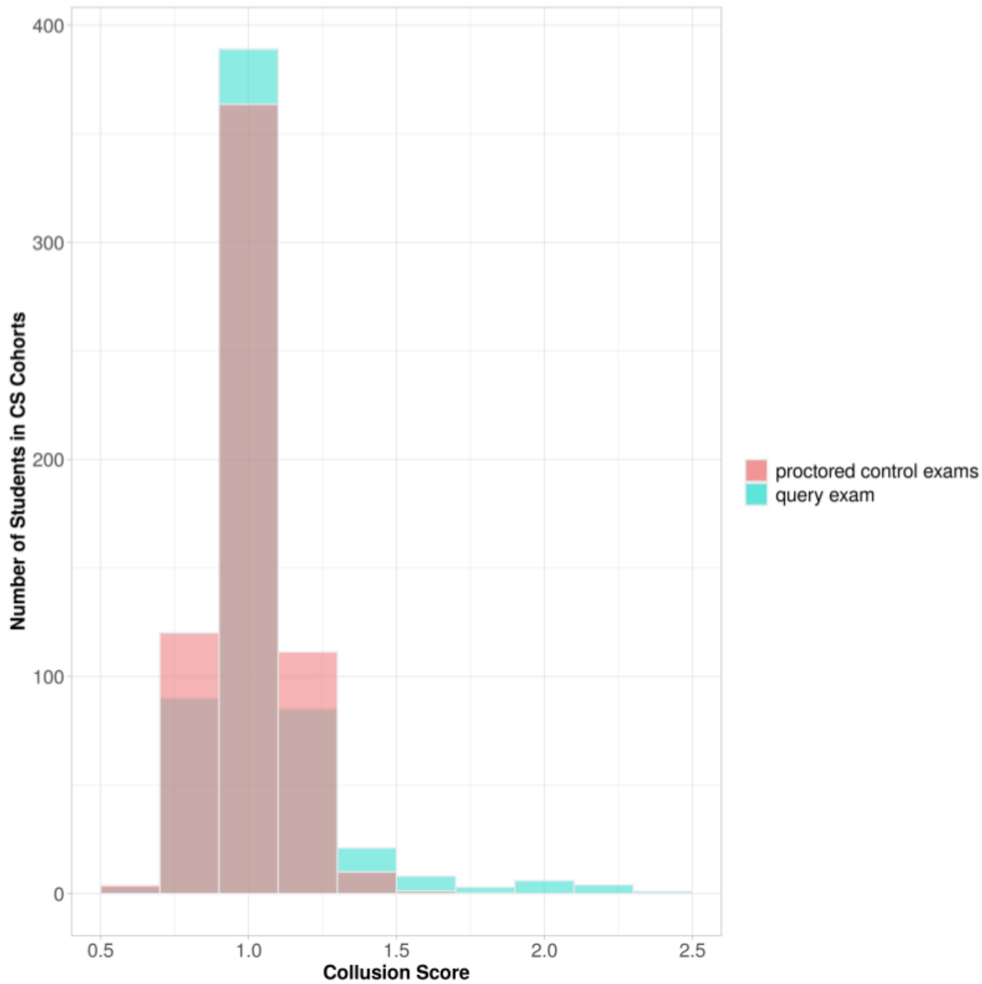


Figure 9.5: Distribution of collusion scores on an unproctored remote exam (CS 161, Spring 2020). Note the presence of high collusion scores in the tail of the distribution.

engage in significant collusion throughout the exam.

In Figure 9.7, the number of colluding groups per exam and the number of colluding students per exam both show that unproctored exams have a higher rate of collusion than in-person exams or proctored exams, as expected. A more surprising result is that the number of colluding students on remote proctored exams is comparable to the number of colluding students on in-person exams, suggesting that remote proctored exams can have exam security comparable to that of in-person exams.

Because Q-SID may have false positives, it might be worth focusing on the number of colluding students with a collusion score over 2.0 (Figure 9.8). With this restriction, the difference between remote proctored exams and in-person exams becomes more pronounced,

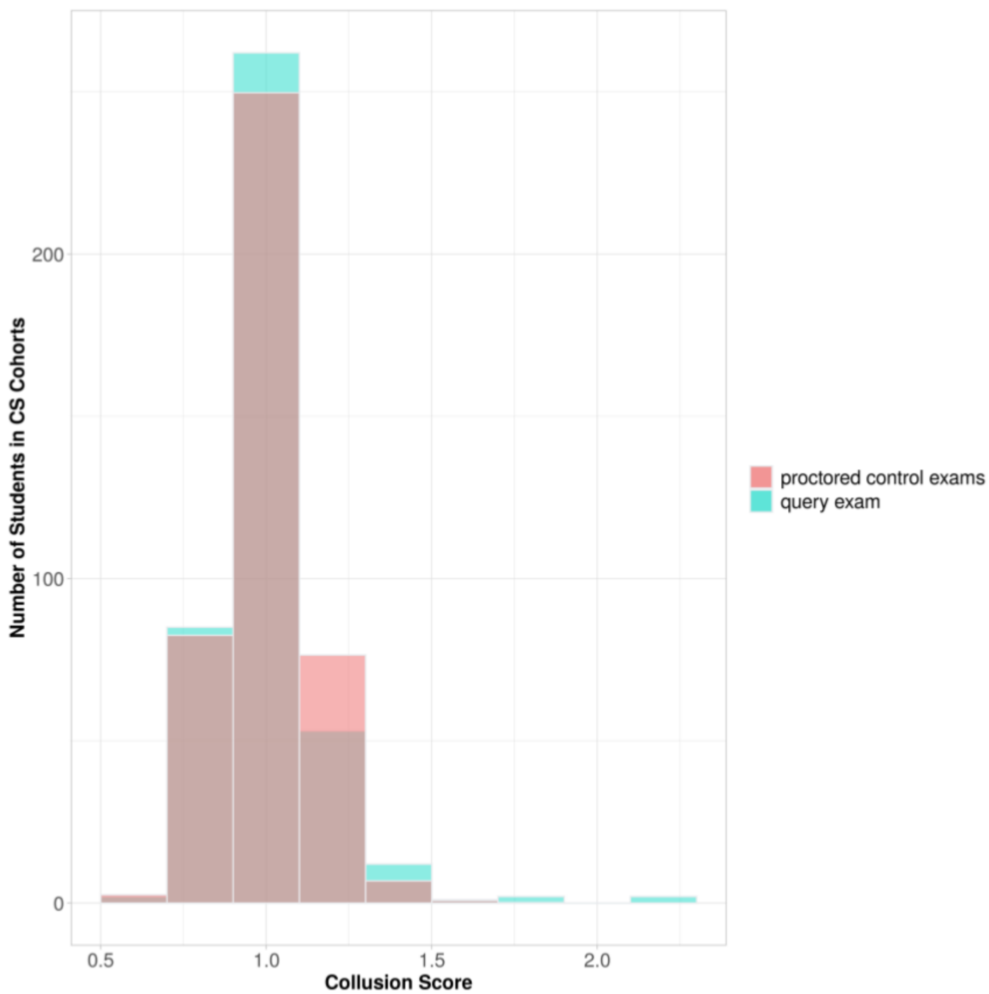


Figure 9.6: Distribution of collusion scores on a proctored remote exam (CS 161, Fall 2020). There are a few instances of high collusion scores, but fewer than in an unproctored exam.

though the number of colluding students is relatively small and likely subject to variance between semesters.

Another way to measure collusion rates is to normalize by class size to obtain the number of colluding students per 100 students (Figure 9.9. This is again subject to variance as a result of small numbers of colluding students each semester.

In Figure 9.10, removing the four exams from summer semesters, where the class sizes were smaller than 100 and subject to higher variance, reveals a similar pattern from earlier: unproctored exams have the highest collusion rates per 100 students, and the collusion rates for remote proctored exams are more similar to those of in-person exams. However, normalizing by class size does show a higher rate of collusion in remote proctored exams.

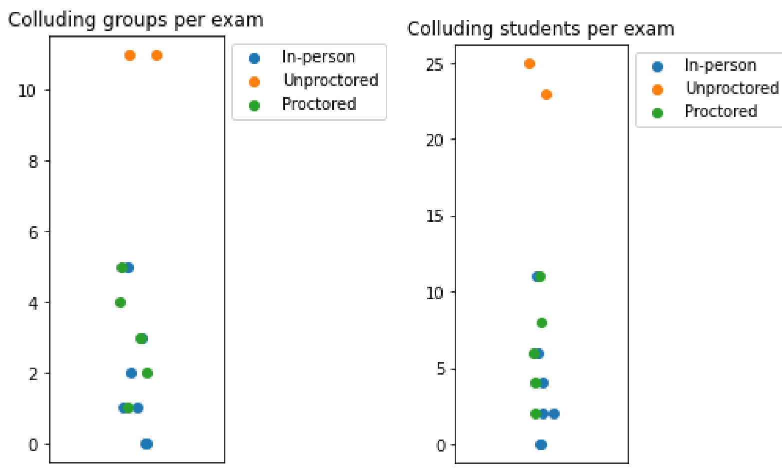


Figure 9.7: Number of colluding groups and number of colluding students for each exam in the dataset, grouped by whether the exam was in-person, unproctored, or proctored.

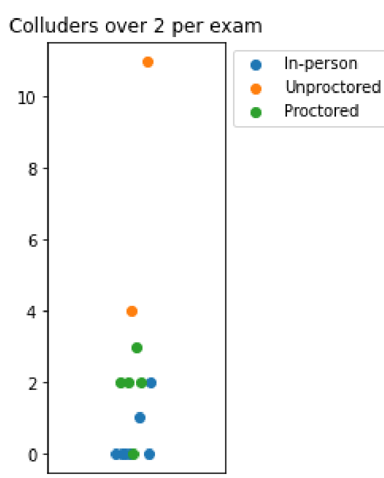


Figure 9.8: Number of number of colluding students with collusion score over 2.0 for each exam in the dataset, grouped by whether the exam was in-person, unproctored, or proctored.

Q-SID can also be used across multiple exams in the same semester to provide stronger evidence of collusion happening on multiple exams. One simple approach to this is looking for groups of colluding students who are flagged by Q-SID independently multiple times on different exams. This occurred most often in the two unproctored exams, where three groups were flagged in both exams. Interestingly, one of these groups was also flagged in the in-person exam in the same semester, suggesting that this group colluded even before

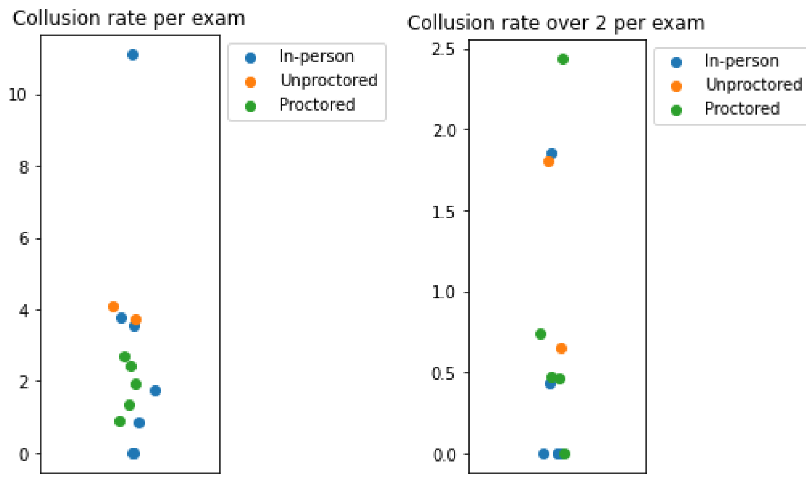


Figure 9.9: Rate of colluding students per 100 students and rate of colluding students with collusion score over 2 (per 100 students) for each exam in the dataset, grouped by whether the exam was in-person, unproctored, or proctored.

remote exams were introduced that semester. Another group was flagged in all 3 exams in a fully in-person semester, which suggests some collusion method on an in-person exam that was exploited (perhaps as simple as sitting near each other and copying an exam).

Finally, an interesting phenomenon from these Q-SID results is a slightly negative correlation between test score and collusion score. In other words, students who score lower on the exam tend to have higher collusion scores, which at first glance seems counterintuitive to the idea that students who successfully collude will end up scoring higher on the exam. One possible explanation for this is the methodology of using only question scores (and not student answers) to detect collusion. If two non-colluding students score poorly on the exam, both students are likely to have answered questions incorrectly, which would show up as a low or zero score for that question. Even though the incorrect answers may be completely different, Q-SID might detect the shared score as evidence of collusion, leading to higher collusion scores and possible false positives. A few concrete examples of this phenomenon resulting in a false positive are shown in the next case study.

In summary, these Q-SID results corroborate related work about remote exams, even when at large scale, which makes it a promising tool for cheat detection in the future. However, these results are not comprehensive because some of these collusion cases were not investigated in their original semester, making it unclear which cases are false positives. Additionally, the small sample size of collusion cases, although good for exam integrity, also makes it impractical to extrapolate any statistically significant results. Future work in Q-SID on large-scale exams can take a more disciplined experimental approach to measuring its effectiveness at detecting collusion.

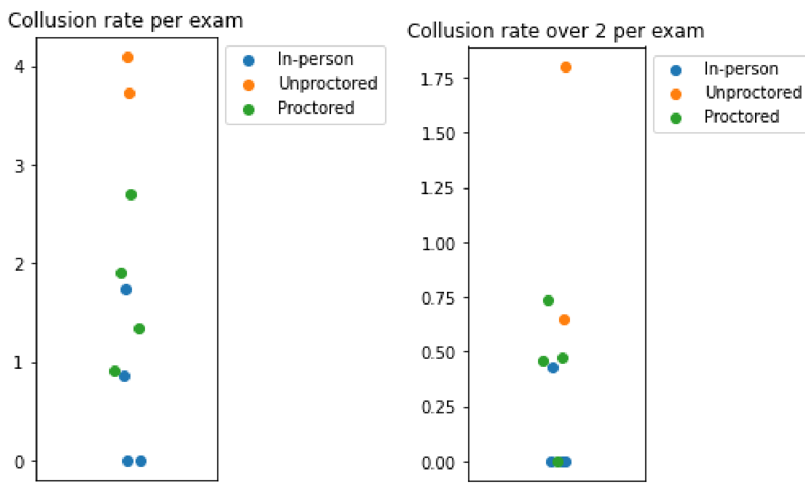


Figure 9.10: Rate of colluding students per 100 students and rate of colluding students with collusion score over 2 (per 100 students) only for exams in the dataset with over 100 students, grouped by whether the exam was in-person, unproctored, or proctored.

9.7 Q-SID Case Study: CS 61C

In this case study, Q-SID was actively deployed during a semester with hybrid exams to detect collusion. The hybrid setup is particularly useful for this study because it allows for comparisons between the same exam delivered in-person and remotely. Additionally, the active deployment of Q-SID provides data about which cases were false positives and which were confirmed cheating cases. Individual cases can be examined for other patterns of collusion that could be used to improve Q-SID or develop alternate methods of collusion detection in the future.

Across the midterm and final, Q-SID detected no cases of collusion in either in-person exam and one pair of colluding students on the remote midterm. On the remote final, Q-SID detected three pairs of students and one large group of 7 colluding students. The colluding pair from the midterm is included in this large group of 7 colluding students. This likely corroborates the theory from the previous case study, that collusion is rare on an in-person proctored exam.

Because of the likelihood of false positives, it is important to analyze each submission by hand to look for further evidence of cheating. When manually looking through Group 1 on the final (collusion score 1.92), we saw that the two students have the same answers on almost every question, a very strong indicator of collusion. Not all matching answers are equally informative, though: some easier questions were correctly answered by more than 90% of the class, so a pair of matching correct answers does not provide the strongest evidence of cheating. Matching incorrect answers are often more useful, as they are less likely to happen by chance. More generally, matching uncommon answers provide stronger evidence of cheating. This also means that some question types are more valuable than others for detecting cheating: for example, in a multiple-choice question with 4 answer choices, two students randomly guessing will have matching answers 25% of the time. Some examples of more valuable matching answers are described below.

Part of a question involved students counting out numbers in hex and writing out hex numbers in a short-answer box. Most students counted `0x00`, `0x01`, `0x02`, `0x03`, etc. in hex, but about a dozen students counted `0x00`, `0x01`, `0x10`, `0x11`, etc. in binary instead. This pair of students both made the same unusual mistake of counting in binary, which was a strong indicator of collusion to us. Another valuable piece of evidence was a fill-in-the-blank coding question, where this pair of students had an answer that was unique to them out of all 285 submissions.

This method of hand-checking answers can probably be automated, especially if student answers consistently follow a certain format (e.g. multiple-choice, numerical answers, short answers with a standard format). This is especially viable on remote exams, where typed answers can be automatically processed (unlike handwritten answers). This method should place higher weight on matching answers that are more unusual, although some experimentation would be required to determine the best weighting algorithm.

Another strong argument in favor of cheating for this pair of students was a clear pattern of intentionally making the same answers look superficially different between the two submissions. Some examples include: equivalent numerical expressions (e.g. 100/19 and 1/0.19), including and leaving out the units (e.g. 5 nanoseconds and 5), changing argument order in a coding question (e.g. a+b and b+a), and changing syntax in a coding question (e.g. a+b without spaces and a + b with spaces). These superficial changes were present on almost every short-answer question, even though on two randomly-chosen exams, one would expect at least some answers to be formatted the same. One particularly egregious change was a question where students needed to print out 20 bytes of garbage data (any 20-byte string was acceptable). Most students chose to print `"A"*20`, because this was the example shown in the question, but one student in the pair deliberately chose to print `"B"*20`. Combined with the other evidence, including the fact that no other student chose to print `"B"*20`, we interpreted this as an attempt to throw off cheat detection mechanisms.

Although this evidence is convincing and helped us build a case on this pair of students, it seems more difficult to build an algorithm that specifically detects this kind of cheat detection circumvention. In fact, the best method for detecting students that try these evasion methods might be Q-SID, since Q-SID only views the scores resulting from student answers and disregards the actual content of student answers. Our intuition with this pair of students turned out to be correct, as they both admitted to collusion at the end of the semester.

Another case worth examining in these results is a pair of exams flagged by Q-SID that was quickly dismissed as a false positive after manual inspection. This quickly became apparent by noting that most answers were different, but incorrect. Because Q-SID only processes question scores and not the actual answers, it may be erroneously using these shared incorrect answers as evidence of collusion. It might be possible to counteract these false positives by placing less emphasis on low or zero scores when looking for collusion, but as described in the previous example, shared incorrect answers are often the strongest indicator of collusion, so this modification might lead to more false negatives. A system that combines Q-SID's answer-agnostic analysis with a more answer-oriented analysis that flags unusual matching answers has strong potential for effectively detecting collusion with minimal false positives. (It should be noted that false positives are probably preferable to false negatives when running cheat detection, though.)

9.8 Detection: Temporal Analysis

Unlike in-person exams, online exam tools can record how students' answers change over time with auto-saved answers, keylogging tools, or submission histories. Analyzing how students' answers change over time is a method for online cheating detection that would not be possible on an in-person exam.

A common form of online cheating involves copy-pasting answers from a friend or some other disallowed online source. Although students may edit their answer after copy-pasting to avoid detection, intermediate logging can be used to detect copied answers. This can be used in conjunction with other prevention and detection strategies such as cheat traps. For example, a student may copy an answer from a friend with a different keyword, and then change the keywords to match their version of the exam. Although their final submission will show no anomalies, the intermediate version provides evidence that they initially copied this answer from someone else's exam.

Temporal analysis combined with cheat traps detected the greatest number of cheating cases, both during the academic year and during the summer. For example, in one particularly remarkable incident, keylogs showed evidence of student A providing the answer to student B, followed by student B correcting the cheat-trap to their own version. Later, student B corrected a mistake in their answer that student A then corrected, demonstrating bidirectional collaboration between the two students.

Temporal analysis can also be used to correlate students who are working through the exam at the same time. Combined with exam randomization, a pair of students filling out the exam out of order to answer the same questions at the same time may be working on the exam together. For example, while conducting detection with Q-SID, it helped to consult keylogs to see if two unusual answers were entered at similar times. Further research is needed to determine the effectiveness and false positive rate of this type of analysis, and whether this analysis could be automated.

The strategy of comparing similar exams can also be extended to include intermediate submissions in order to have a more complete picture of students who may have had suspiciously similar exams at any point during the exam.

Chapter 10

Conclusion and Future Work

Over the past two years, EECS instructors and students worked together to develop a wide range of policies, platforms, and tools to support remote and hybrid exams at scales of up to a few thousand students. This report has documented these developments, as well as the results and experiences from the remote exam space as a whole.

It is worth noting that all the results in this paper come from real-world observational data collected during the pandemic. Although this data contains insights about the effectiveness of various remote exam strategies and points toward potential future projects, the data is not as rigorous as data collected from a controlled experiment. Future work might focus on devising more controlled experiments to test the hypotheses presented in this report. For example: all else equal, do students score lower on remote exams than in-person exams, and if so, what are the root causes? All else equal, do students submit more clarifications in a remote exam than an in-person exam? More rigorous experiments can also be conducted on the cheat detection methods proposed in this report to get a better sense of their effectiveness and their shortcomings.

It is also worth considering how these policies extrapolate to other computer science programs, which may have different scales and needs compared to the Berkeley EECS department. Policies that worked well for EECS, particularly because of its large class sizes and undergraduate TA population, may not work as well for smaller class sizes that are managed more centrally by an instructor.

A longer-term project can build on the hypotheses and insights from this report to develop a more robust remote exam delivery and proctoring tool. Many of the tools used in EECS during the pandemic were temporary and not meant for long-term use, such as CS 61A Exam Tool for exam delivery and Zoom for exam proctoring. An integrated remote exam system could combine exam delivery and proctoring into a single integrated in-browser application for student and staff convenience. Such a system can also incorporate the grading and cheating detection methods described in this report to create one seamless, comprehensive remote exam platform.

An important takeaway from this report is that although remote exams are already being phased out as schools return to in-person instructions, the developments in remote learning have had immediate benefits for disadvantaged students. We hope that others will continue to build on our findings about remote and hybrid exams in order to preserve their unique benefits and create a more equitable learning environment. Although this report may seem like a historical document describing a temporary solution to a temporary problem, we hope that it is instead the start of a larger conversation about how we can leverage the power of remote learning tools to create a learning environment that is flexible and adaptable in the face of real-world events and accessible to all students.

Appendix A

Remote Exam Student-Facing Policy

The midterm exam is on $DATE, $TIME ($TIMEZONE). If you cannot make this time, please fill out this conflict form.

If you'd prefer to upload a local recording instead of using Zoom, or you'd prefer to record a screen share and webcam feed instead, please reach out privately.

If you feel uncomfortable with proctoring, please reach out so you can discuss alternatives with an instructor.

A.1 Before the exam:

- Prepare cheat sheets (unlimited, handwritten). Handwritten tablet notes can be printed and used. If you have handwritten tablet notes but no printer access, let us know privately.
- Download Zoom on your smartphone and experiment with a video recording setup (see below). You can see examples of setups from course staff here (see Figure 3.1). Please reach out privately if you are having setup difficulties.

A.2 On the day of the exam:

- Join the Zoom meeting and position your smartphone so that we can clearly see your computer/tablet screen, and if possible, both of your hands. The exam won't start until everyone's in the Zoom room, so you may have to wait a few minutes.
- Open the exam on $PLATFORM. When you're ready to start the exam and the proctor has announced that the exam has started, check the clarifications page for the decryption password.

A.3 During the exam:

- Only the exam page should be on your computer screen.
- Submit clarification requests through the exam platform. If your question is answered, it will be displayed as an announcement. Please check the announcements periodically during the exam.

A.4 Technical issues:

- Don't worry if your video feed disconnects briefly during the exam.
- If you encounter significant technical problems, don't worry about video proctoring and focus on finishing the exam.
- We may ask some students to take a short verbal exam after the exam and explain how to solve one or more problems that are similar to an exam question they got right.
- If you encounter Internet problems, write your answers locally and send them to $EMAIL as soon as the exam is over.
- If you need to use the bathroom, just leave the video feed on while you're away.

A.5 Privacy policy:

- Course staff will not save any images or recordings from the proctoring session after the exam is over.
- Course staff will respect your privacy and not disclose any information from the proctoring session after the exam, except in cases of academic dishonesty.
- Every case of potential academic dishonesty will be manually reviewed, and you'll be able to discuss the situation with the instructor.
- You have the option to join the meeting with your SID as your display name for anonymity. We will tell you who your proctor is before the exam. You have the option to switch proctors for any reason: please reach out privately before the exam if you would like to switch proctors.

Appendix B

Hybrid Exam Student-Facing Policy

The final exam will be held on $DATE, $TIME ($TIMEZONE).

B.1 Final Exam

If you would like to request an online exam for any reason or cannot make this time, please fill out the final exam alterations form by $DATE. If you want to request an adjustment for your exam, such as a left-handed desk or a seat in a specific building, please also let us know through the form. If you encounter extenuating circumstances after the form deadline, you may still fill out the form, though we can't guarantee any last-minute adjustments.

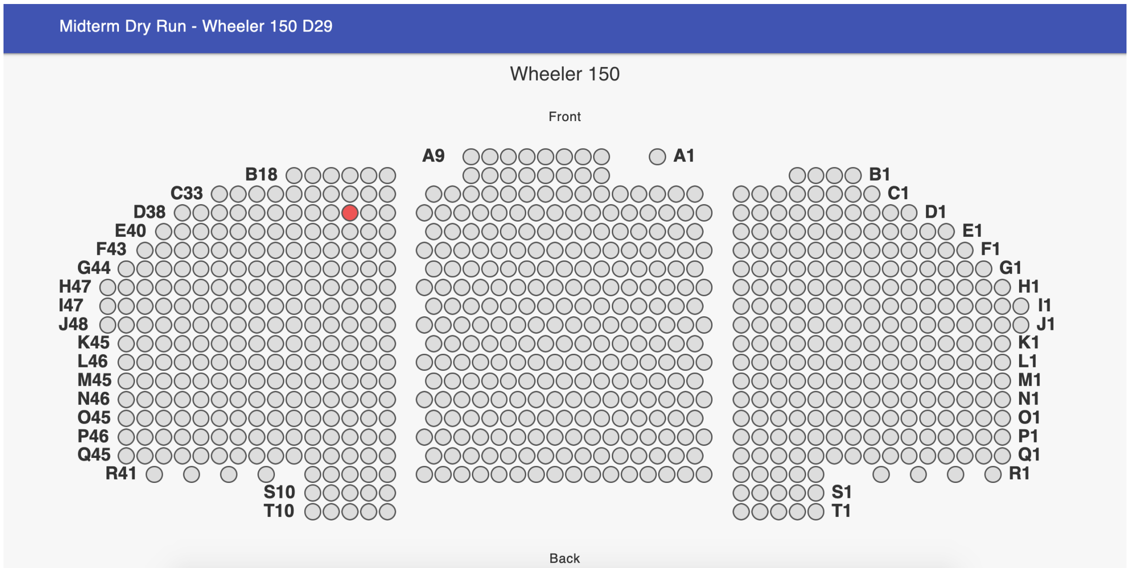
Final Resources

The exam is closed-book, closed-note, closed-internet, with the following exceptions:

- You can prepare and use up to 10 two-sided cheat sheets (20 sides) for the final exam. You can reuse your midterm cheat sheets.
  - Cheat sheets must be handwritten. Digitally handwritten notes may be printed and used. If you're taking the exam online and have digitally handwritten notes but no printer access, let us know.
- On the exam, you will be provided with the reference card.

B.2 In-Person Exam Logistics

- In-person exams will be conducted in assigned exam rooms. You will receive confirmation of your exam room and seat number the night before the exam. Here's a screenshot of the seating tool.



On the Day of the Exam

- If you're not feeling well, please submit the alterations form ASAP so we can move you to an online exam.
- Please bring:
  - Mask
  - Pencil/erasers or pen (black or blue ink)
  - Student ID card
  - Cheat sheets (if you have any)
- Please **try to get to your room early**, so that we can start on time.
- We'll give you a paper copy of the exam on which you can write your answers, and a separate appendix packet.

During the Exam

- First in-person exam? Don't worry, proctors will announce everything you will need to do.
- If you finish before the last 10 minutes of the exam, you can turn your exam in and leave early.

—-

B.3 Online Exam Logistics

- Online exams will use Zoom proctoring. You will receive a Zoom link the night before the exam.
  - If you want to know who your proctor is, please reach out to course staff.
  - If you feel uncomfortable with proctoring, please reach out so you can discuss alternatives with an instructor.

On the Day of the Exam

- Join the Zoom meeting on the computer you will be using and share your screen.
- Set up a camera (join Zoom through a separate device or turn on webcam video) showing your workspace. You can see examples of setups from course staff here (see Figure 3.1).
- Optionally, if you want, you can rename yourself to your SID instead of your name.
- On your computer, open up Examtool and select your exam (e.g. `cs61c-su22-final-5pm`) from the drop-down list (the selection will appear a few hours before the exam).
- The decryption password will be posted on Examtool at 5:10 PM for the 5 PM exam.

During the Exam

- Only the Examtool page and the reference card should be on your computer screen. If you don't have access to a printer for your cheat sheets, please let us know.
- Please submit clarification requests only through Examtool.
- When you're done, feel free to close Examtool (your exam will be autosaved) then leave the call.

Technical Issues

- Don't worry if your video feed disconnects briefly during the exam.
- If you encounter significant technical problems, don't worry about video proctoring and focus on finishing the exam.
- We may ask some students to take a short verbal exam after the exam and explain how to solve one or more problems that are similar to an exam question they got right.
- If you encounter Internet problems, write your answers locally and send them to $EMAIL as soon as the exam is over.

- If you need to use the bathroom, just leave the video feed on while you're away.

Bibliography

[1] *Accommodation Terms*. 2022. URL: https://dsp.berkeley.edu/faculty/proctoring/your-department.

[2] A. Paul Alivisatos and Oliver O'Reilly. *Remote Proctoring Policy for Summer Sessions and Fall Semester*. 2020. URL: https://calmessages.berkeley.edu/archives/message/75444.

[3] Paul Alvisatos. *Executive order regarding exam proctoring*. 2020. URL: https://calmessages.berkeley.edu/archives/message/73444.

[4] William R Balch. "Item order affects performance on multiple-choice exams". In: *Teaching of Psychology* 16.2 (1989), pp. 75–77.

[5] Eren Bilen and Alexander Matros. "Online cheating amid COVID-19". In: *Journal of Economic Behavior & Organization* 182 (2021), pp. 196–211. ISSN: 0167-2681. DOI: https://doi.org/10.1016/j.jebo.2020.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0167268120304510.

[6] Nicholas Casey. *College Made Them Feel Equal. The Virus Exposed How Unequal Their Lives Are.* Apr. 2020. URL: https://www.nytimes.com/2020/04/04/us/politics/coronavirus-zoom-college-classes.html.

[7] Binglin Chen, Matthew West, and Craig 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*. L@S '18. London, United Kingdom: Association for Computing Machinery, 2018. ISBN: 9781450358866. DOI: 10.1145/3231644.3231664. URL: https://doi.org/10.1145/3231644.3231664.

[8] Binglin Chen et al. "Learning to Cheat: Quantifying Changes in Score Advantage of Unproctored Assessments Over Time". In: *Proceedings of the Seventh ACM Conference on Learning @ Scale*. L@S '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 197–206. ISBN: 9781450379519. DOI: 10.1145/3386527.3405925.

[9] Weiqin Chen. "Students with Disabilities and Digital Accessibility in Higher Education under COVID-19". In: *Proceedings of the 29th International Conference on Computers in Education*. Asia-Pacific Society for Computers in Education, 2021, pp. 656–662.

[10] Carol Christ. *URGENT: New campus actions regarding COVID-19*. 2020. URL: https://calmessages.berkeley.edu/archives/message/72925.

[11] Carol Christ and A. Paul Alivisatos. *UC system plans resume in-person instruction by fall 2021*. 2021. URL: https://news.berkeley.edu/2021/01/11/uc-system-plans-resume-in-person-instruction-by-fall-2021/.

[12] Carol Christ and A. Paul Alivisatos. *Update on fall semester planning*. 2021. URL: https://calmessages.berkeley.edu/archives/message/83924.

[13] GR Cluskey Jr, Craig R Ehlen, and Mitchell H Raiborn. "Thwarting online exam cheating without proctor supervision". In: *Journal of Academic and Business Ethics* 4.1 (2011), pp. 1–7.

[14] Seife Dendir and R. Stockton Maxwell. "Cheating in online courses: Evidence from online proctoring". In: *Computers in Human Behavior Reports* 2 (2020). ISSN: 2451-9588. DOI: 10.1016/j.chbr.2020.100033. URL: https://www.sciencedirect.com/science/article/pii/S2451958820300336.

[15] Jarret Dyer, Heidi Pettyjohn, and Steve Saladin. "Academic Dishonesty and Testing: How Student Beliefs and Test Settings Impact Decisions to Cheat". In: *Journal of the National College Testing Association* 4 (2020).

[16] Lina Elsalem et al. "Stress and behavioral changes with remote E-exams during the Covid-19 pandemic: A cross-sectional study among undergraduates of medical sciences". In: *Annals of Medicine and Surgery* 60 (2020), pp. 271–279. ISSN: 2049-0801. DOI: 10.1016/j.amsu.2020.10.058. URL: https://doi.org/10.1016/j.amsu.2020.10.058.

[17] *Final exam using gradescope for 1200 students crashed HARD tonight because...they weren't prepared for more than 300 stus in Canada?! I. Don't. Understand. Now take-home exam, stus super-stressed and I will NEVER answer all the emails.* 2020. URL: https://twitter.com/funnyfishes/status/1253552115466682369.

[18] Edward Gehringer. "Online Vs. On Paper Exams". In: *Proceedings of the 2010 American Society for Engineering Education Annual Conference & Exposition*. 2010, pp. 15.927.1–15.927.19. DOI: 10.18260/1-2--16349. URL: https://peer.asee.org/online-vs-on-paper-exams.

[19] Gabriel Geiger. *Students Are Easily Cheating 'State-of-the-Art' Test Proctoring Tech*. Mar. 2021. URL: https://www.vice.com/en/article/3an98j/students-are-easily-cheating-state-of-the-art-test-proctoring-tech.

[20] Afshin Gharib, William Phillips, and Noelle Mathew. "Cheat Sheet or Open-Book? A Comparison of the Effects of Exam Types on Performance, Retention, and Anxiety." In: *Online Submission* 2.8 (2012), pp. 469–478.

[21] Oskar R. Harmon and James Lambrinos. "Are Online Exams an Invitation to Cheat?" In: *The Journal of Economic Education* 39.2 (2008), pp. 116–125. DOI: 10.3200/JECE.39.2.116-125. URL: https://doi.org/10.3200/JECE.39.2.116-125.

[22] Drew Harwell. *Cheating-detection companies made millions during the pandemic. Now students are fighting back.* Nov. 2020. URL: `https://www.washingtonpost.com/technology/2020/11/12/test-monitoring-student-revolt/`.

[23] Rebecca Heilweil. *Paranoia about cheating is making online education terrible for everyone.* May 2020. URL: `https://www.vox.com/recode/2020/5/4/21241062/schools-cheating-proctorio-artificial-intelligence`.

[24] Benjamin Herold. "The disparities in remote learning under coronavirus (in charts)". In: *Education Week* 10 (2020).

[25] *Instructor guidance (fall 2021).* 2021. URL: `https://docs.google.com/document/d/1FUY6LYO-QSbtSo0k4V37XF00ijBw7iBMxGpkesFGy3g`.

[26] Anya Kamenetz. *A Growing Number Of U.S. Colleges Cancel Classes Amid Coronavirus Fears.* Mar. 2020. URL: `https://www.npr.org/2020/03/09/813750481/more-than-20-colleges-cancel-in-person-classes-in-response-to-coronavirus`.

[27] Chula G. King, Jr. Roger W. Guyette, and Chris Piotrowski. "Online Exams and Cheating: An Empirical Analysis of Business Students' Views". In: *Journal of Educators Online* 6.1 (2009). URL: `https://eric.ed.gov/?id=EJ904058`.

[28] Catherine P. Koshland, Jenna Johnson-Hanks, and Becca Lopez. *Important Reminders: Academic Integrity and Honor Code.* 2020. URL: `https://calmessages.berkeley.edu/archives/message/80688`.

[29] Thomas Lancaster and Codrin Cotarlan. "Contract cheating by STEM students through a file sharing website: a Covid-19 pandemic perspective". In: *International Journal for Educational Integrity* 17.3 (2021). DOI: `s40979-021-00070-0`. URL: `https://doi.org/10.1007/s40979-021-00070-0`.

[30] Kevin Lin. "A Berkeley View of Teaching CS at Scale". MA thesis. EECS Department, University of California, Berkeley, May 2019. URL: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-99.html`.

[31] Frank M LoSchiavo and Mark A Shatz. "The impact of an honor code on cheating in online courses". In: *MERLOT Journal of online Learning and Teaching* 7.2 (2011).

[32] Donald L McCabe. "Cheating among college and university students: A North American perspective". In: *International Journal for Educational Integrity* 1.1 (2005).

[33] Donald L McCabe, Linda Klebe Trevino, and Kenneth D Butterfield. "Academic integrity in honor code and non-honor code environments: A qualitative investigation". In: *The Journal of Higher Education* 70.2 (1999), pp. 211–234.

[34] Donald L. McCabe. "The Influence of Situational Ethics on Cheating Among College Students". In: *Sociological Inquiry* 62.3 (1992), pp. 365–374. DOI: `10.1111/j.1475-682X.1992.tb00287.x`. URL: `https://doi.org/10.1111/j.1475-682X.1992.tb00287.x`.

[35] Darlene L. Neely, Frederick J. Springston, and Stewart J. H. McCann. "Does Item Order Affect Performance on Multiple-Choice Exams?" In: *Teaching of Psychology* 21.1 (1994), pp. 44–45. DOI: 10.1207/s15328023top2101_10. eprint: https://doi.org/10.1207/s15328023top2101_10. URL: https://doi.org/10.1207/s15328023top2101%5C_10.

[36] Oliver M. O'Reilly and Ronald C. Cohen. *Fall 2021 Remote Proctoring Policy*. 2021. URL: https://calmessages.berkeley.edu/archives/message/87964.

[37] Sandra Gudiño Paredes, Felipe de Jesús Jasso Peña, and Juana María de La Fuente Alcazar. "Remote proctored exams: Integrity assurance in online education?" In: *Distance Education* 42.2 (2021), pp. 200–218. DOI: 10.1080/01587919.2021.1910495. URL: https://doi.org/10.1080/01587919.2021.1910495.

[38] *Religious Holidays & Religious Creed Policy*. 2010. URL: https://registrar.berkeley.edu/calendar/.

[39] Mihaela Sabin, Karen H. Jin, and Adrienne Smith. "Oral Exams in Shift to Remote Learning". In: *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 666–672. ISBN: 9781450380621. URL: https://doi.org/10.1145/3408877.3432511.

[40] Carla van de Sande and Xuetao Lu. "Perceptions of Cheating on In Person and Online Mathematics Examinations". In: *International Journal of Arts Humanities and Social Sciences* 3.9 (2018), pp. 28–35.

[41] Richard B. Schultz and Michael N. DeMers. "Transitioning from Emergency Remote Learning to Deep Online Learning Experiences in Geography Education". In: *Journal of Geography* 119.5 (2020), pp. 142–146. DOI: 10.1080/00221341.2020.1813791. eprint: https://doi.org/10.1080/00221341.2020.1813791. URL: https://doi.org/10.1080/00221341.2020.1813791.

[42] Berkeley Academic Senate. *Frequently Asked Questions - Instructor FAQ fall 2021*. 2021. URL: https://academic-senate.berkeley.edu/faq/instructor-faq-fall-2021#t38n670.

[43] Derar Serhan. "Transitioning from face-to-face to remote learning: Students' attitudes and perceptions of using Zoom during COVID-19 pandemic". In: *International Journal of Technology in Education and Science* 4.4 (2020), pp. 335–342.

[44] Yang Song and David Thuente. "A quantitative case study in engineering of the efficacy of quality cheat-sheets". In: *2015 IEEE Frontiers in Education Conference (FIE)*. 2015, pp. 1–7. DOI: 10.1109/FIE.2015.7344082.

[45] Shea Swauger. "Our Bodies Encoded: Algorithmic Test Proctoring in Higher Education". In: *Hybrid Pedagogy*. 2020.

[46] *Syllabus*. 2019. URL: https://cs186berkeley.net/fa19/syllabus.

[47] *Traveling & Exam Proctoring*. 2017. URL: https://asc.berkeley.edu/resources/students/academic-advising-resources/traveling-exam-proctoring.

[48] Matthew West, Geoffrey L Herman, and Craig Zilles. "Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning". In: *2015 ASEE Annual Conference & Exposition*. 2015, pp. 26–1238.

[49] Daniel Woldeab and Thomas Brothen. "21st Century Assessment: Online Proctoring, Test Anxiety, and Student Performance". In: *International Journal of E-Learning & Distance Education* 34.1 (2019). URL: https://eric.ed.gov/?id=EJ1227595.

[50] Bojin Yao et al. "Formal Categorization of Variants for Question Generators in Computer-Based Assessments". In: *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2021, p. 1244. ISBN: 9781450380621. URL: https://doi.org/10.1145/3408877.3439683.

[51] Larry Yueli Zhang et al. "A Multi-Course Report on the Experience of Unplanned Online Exams". In: *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 17–23. ISBN: 9781450380621. URL: https://doi.org/10.1145/3408877.3432515.

[52] Sarah Zhang. *This School Year Is Going to Be a Mess—Again*. Apr. 2021. URL: https://www.theatlantic.com/health/archive/2021/08/how-will-delta-affect-schools/619865/.

[53] Craig Zilles et al. "Computerized testing: A vision and initial experiences". In: *2015 ASEE Annual Conference & Exposition*. 2015, pp. 26–387.