Satisfactory for all: supporting mastery learning with human-in-the-loop assessments in a discrete mathematics course

Shao-Heng Ko
shaoheng.ko@duke.edu
Duke University
Durham, North Carolina, USA

Alex Chao
alex.chao@duke.edu
Duke University
Durham, North Carolina, USA

Violet Pang
tong.pang@duke.edu
Duke University
Durham, North Carolina, USA

ABSTRACT

This experience report documents an attempt at embracing the "A's for all" and equitable grading frameworks in an introductory, proof writing-based discrete mathematics course for computer science majors (with $N = 138$ students) at a medium-sized research-oriented university in the US. Unlike in introductory programming contexts, there is so far no reliable automated grading system that gives formative and adaptive feedback supporting the scope of a proof-based discrete mathematics course. We therefore faced the unique challenge of being unable to automate all assessments and directly offer all students unlimited attempts toward mastery.

To address this issue, we adopted a hybrid approach in designing our formative assessments. Using the *Exemplary, Satisfactory, Not Yet,* and *Unassessable* (ESNU) discrete grading model, we required all students to get a Satisfactory or above in every question in every assignment within two rounds of human feedback. Students not meeting the goal after two attempts then consulted with course staff members in one-on-one interactions to get diagnostic feedback at any time at their convenience until the semester ended.

We document our course policy design in detail, then present data that summarizes both the grading outcomes and student sentiments. We also discuss the lessons learned from our initiative and the necessary staff-side management practices that support our design. This report outlines an example of adopting the A's for all and equitable grading framework in a course context where not all contents can be made autogradable.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

A's for all; Equitable Grading; Specifications Grading; Mastery Learning; Discrete Mathematics

ACM Reference Format:
Shao-Heng Ko, Alex Chao, and Violet Pang. 2025. Satisfactory for all: supporting mastery learning with human-in-the-loop assessments in a discrete mathematics course. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701854

1 INTRODUCTION

In recent years, there have been increasing discussions and adoptions of the A's for all [1, 15, 17–19] and Equitable Grading [2, 12, 14, 23, 25] initiatives within the computing education community. A's for all is an initiative/mindset shift that builds on the core idea of shifting the focus of the grading schemes in our classrooms from *fixed time, variable learning* to *fixed learning, variable time* [15], while Equitable Grading focuses on building unbiased grading schemes that support self-regulatory learning [8]. Both approaches can be seen as practical implementations of Bloom's mastery learning model [3]: given a *fixed* line of mastery, every student can achieve that mastery given adequate (and *variable*) time and support resources.

Successful adoption of A's for all conventionally hinges on the automation of all assessments [15] to provide students as much practice as necessary for mastery. While this is a reasonable request in an introductory to programming (CS1) context thanks to autograders [24], automating assessments with formative feedback remains a challenge in many other contexts including mathematical proof-writing [50]. This experience report documents our attempt at adopting A's for all and Equitable Grading practices in an introductory discrete mathematics context *without* being able to fully automate formative assessments.

2 BACKGROUND AND RELATED WORKS

A's for all. An ideal implementation of A's for all allows students to work on acquiring mastery past the official endpoint of the course [4], although instructors usually operate within the limitations of a fixed term (i.e., all work has to be completed before a timepoint to be considered for grading). Garcia *et al.* [15] proposes the following policies to provide students with full agency towards mastery under this scenario:

- **Unlimited attempts:** students should get *as much practice as desired, with immediate and detailed feedback* [15] until mastery. This is often achieved in a CS1 context via randomly generated questions [28, 37] hosted on mastery learning-focused platforms such as PrairieLearn [9, 49]. Instructors also need sufficient administrative support in managing the flexible policies.
- **Absolute grading:** all grading should be based on learning goals, and no normative "curves" (that tie a student's success to any peer's success) should be used.
- **Transparent rubrics:** students should have access to what it takes to achieve mastery. In the context of automated programming assignments, no hidden test cases should be used.
- **Flexible extensions:** students should have a *suggested* deadline on each assignment that reflects the typical pace of the course contents but also be provided an easy means to request extensions

to cope with difficulties or unforeseen circumstances, ideally through an automated process [36, 44]. Instructors must also strike a balance between flexibility and procrastination [4, 16, 30].
- **Clobbered exams:** exams should be cumulative, so a later exam's higher score can replace an earlier exam's lower score.
- **Easy tracking:** students need a simple and easy-to-use "dashboard" [1] to track what concepts have been mastered, as well as the interdependencies between learning goals.

Equitable Grading. Decker *et al.*'s call to "transform the way the community of CS educators and learners perform, understand, and use assignment grading" [8] suggests adopting ideas from Feldman [12], Nilson [29], and Rapaport [35]'s alternative grading practices:
- **Reduced grading scale** [26, 47]: Replace the traditional, 0-100 points-based grading scales with smaller ones that allow for easier calibration and eliminate points-based negotiations.
- **Encouraged resubmissions** [31, 38]: Allow (and fully encourage) students to revise their work upon previous feedback.
- **Learning outcomes based scale:** Directly tie any grading scale used to the learning outcomes of each assignment.

See Harrington *et al.*'s survey [23] on usage in computing classes.

Automated feedback for mathematical proof writing. Unlike in CS1 contexts [21, 22, 24], no tools have successfully "dealt with grading mathematical proofs using natural language processing" [50]. As a recent benchmark paper shows [32], even when the task is limited to reordering lines into a correct solution in Proof Blocks [33, 34], large language models such as GPT-4 struggle at coming up with correct proofs, let alone *evaluating* student-generated proofs and *giving useful feedback*. Therefore, in the immediate future, any proof-writing course still inevitably needs to rely on humans to provide personalized feedback to students.

Similar courses. We are hardly the first in adopting various A's for all and Equitable Grading designs in a non fully-autogradeable computing course. There have been a few experience reports on discrete mathematics for computer science [43, 46], proof-based pure mathematics [6, 41], and algorithms [5, 39, 48] courses with emphasis on various alternative grading designs. However, we are not aware of any such course that *requires* mastery of *all* core course content from *all students*. We came up with course policy (Section 4.2) and staff-side (Section 4.3) designs to support this ambitious goal, and we believe this unique experience adds to the current documentary of A's for all and Equitable Grading practices.

3 INSTITUTIONAL CONTEXT

Our discrete mathematics course (DM) is hosted at Duke University, a medium-sized, research-oriented, private university in the United States with 15-week semesters. DM is required for computer science majors and has notoriously been known as being *unstable*: 5 different instructors had taught in the past 10 regular semesters before the Spring 2024 semester (this report), and the course content fluctuated drastically among instructors. The primary instructor for the Spring 2024 offering is not one of the 5 previous instructors. This provided us with a great opportunity to adopt new policies without having to worry about disrupting a stable course context.

The Spring 2024 offering had 138 students, and the enrollments for the 10 past regular semesters ranged from 95 to 173 with a mean of 129.1. The course staff consisted of the instructor, a departmental staff member that oversaw course logistics, a graduate *head grader* that oversaw the grading (these are the authors of this experience report), and 24 other TAs (2 graduate and 22 undergraduate).[1]

4 SPRING 2024 COURSE DESIGN

Course content. We split the content into two categories:

Table 1: Core modules topics and formative assignments.

No.	Topic	PrairieLearn	Human-graded
CM1	Logic		X
CM2	Proof Methods	X	X
CM3	Sequences, Recurrences, & Asymp. Notations		X
CM4	Sets, Functions, & Relations	X	X
CM5	Inductions	X	X
CM6	Graph Theory	X	X
CM7	Combinatorics	X	X
CM8	Probability	X	X

- **Core modules** (8 in total; see Table 1) are the topics that *every* student is required to complete. We described them as *"topics that we think **most** computer scientists (i.e., you) need, and/or **most** advanced topics in CS will require."* All core modules were delivered with in-person synchronous class meetings (see Section 4.1).
- **Elective modules** (6) are other topics that *"**some** computer scientists need, and/or **some** advanced topics in CS will require"*; students were not expected to master or even engage with all elective modules (although some did). Delivery modalities varied.

Grading schemes. We used a combination of the *Exemplary, Satisfactory, Not Yet, Unassessable* (ESNU) [26, 48] four-level scale at the item-level and specification grading [23, 29] at the course-level in our grading scheme. While we used the ESNU scale similar to how it has been used in programming [26, 27] and algorithms [48] contexts, we adapted the semantics of the scales into the mathematical proof context in our communication with the students:
- **Exemplary** means a correct proof or work that displays full mastery, where no improvements – either technical or stylistic – are necessary.
- **Satisfactory** means a technically correct proof or work that displays mastery but with potential room for improvements (e.g., logical leaps that could use elaboration, minor abuses of notations, or using more cases than necessary). Any non-proof work that shows full mastery but is numerically incorrect will also be categorized as Satisfactory.
- **Not Yet** means the work is technically incorrect or incomplete, does not display full mastery, and thus needs revisions.
- **Unassessable** means the work is missing or too incomplete for the staff to give concrete feedback.

At the beginning of the semester, we told the students that the difference between Exemplary and Satisfactory *at the question level* is purely nominal and does not factor into any grade aggregation. In other words, neither Satisfactory nor Exemplary works have to be resubmitted, and the distinction lies in that students should consult any feedback the course staff attached to Satisfactory work. We hereon refer to "Satisfactory or Exemplary" as "S+" for brevity.

[1]While the staff head count seems high, most undergraduate TAs only worked for 6 hours per week on average, including 1.5-2 hours per week of training. The amount of work hours we used was in line with past DM offerings at our institution.

Students' final letter grade was a non-linear aggregate of (1) the number of core modules completed, (2) the number of elective modules completed, and (3) their performances in the three in-person paper exams. *The completion of all core modules was required for all letter grades C- and above*, while D-range letter grades allowed for missing one core module. Similar to many other core courses at our institution, students had two chances at taking each exam, where the second attempt for all exams was administered at the end of the semester in lieu of a final exam.

Due to space constraints and the less stable context of elective modules, we will devote the remaining parts of this report to the core modules. All terms such as "modules", "assignments", etc. from now on pertain to only the core modules; please refer to our course website [11] for all details not included in this report.

4.1 Core modules dynamics

Readings and quizzes. The modules adopted a *partially flipped* learning model. Students were assigned textbook chapters to consume prior to each module's first class meeting. Each module had a lightweight conceptual quiz built in the Learning Management System for students to self-evaluate their proficiency of concepts. These quizzes had unlimited *throttled* attempts[2] with binary feedback (correct/incorrect), and students were allowed (and encouraged) to discuss the quiz contents on the discussion forum with peers. Students were made aware that both the readings and the quizzes were due the night before each module's first class meeting in a *suggested* fashion without any penalty, and they were welcome to continue using both as learning materials throughout the semester.

Class meetings. The instructor approached the (in-person and synchronous) class meetings assuming all students present had engaged with the reading and quizzes.During each module's (2-3) class meetings, the instructor motivated the need to learn the module topic, compared and contrasted between different textbooks' contents, and (in later modules) explicitly referred back to earlier modules and formed connections between the concepts. The rest of in-class time was allocated to peer instructions [7, 45] as well as covering additional material beyond required readings.

Recitation sessions. Students also engaged in weekly TA-led recitation sessions (our offering had 5 such sessions, each led by 2-3 TAs with 20-30 students). All recitation sessions ran the same problem-solving-based material that built on class meeting content. While attendance in recitations was historically part of students' grades in previous iterations of DM, we felt this contradicted the spirit of mastery learning as it based students' grades on things irrelevant to what students have learned and promoted participation for the sake of earning points. We thus required the *work* instead of *attendance* in these sessions: students were to either attend the session (so the work was assumed completed) or individually submit all the work (in which case a staff member performed a cursory check).

Autograded assignments. Six of the eight modules had an autograded component in the corresponding assignment (see Table 1). We hosted the autograded part on the PrairieLearn (PL) [49] platform with unlimited attempts and only *suggested* due dates. This part included Proof Block [34] style questions (to ease students into proof writing) and randomized numerical questions (to provide more practice with combinatorics and probability). We shifted a part of (historically exclusively human-graded) assignments to autograding platforms mainly to help reduce staff workload (see Section 4.3) and save the human resources for multiple rounds of feedback. We also integrated PL into the feedback process of the human-graded component in one module (see Section 5.1).

Human-graded assignments. Each module had an exclusively human-graded assignment component administered in Gradescope [20], with most questions being one of two non-autogradeable flavors: (1) close-ended asking students to prove a claim; (2) semi-open-ended asking for an example/counterexample satisfying certain criteria.

We encouraged the students to work in pairs[3] on these human-graded assignments provided that they did not work with the same partner in consecutive assignments. This policy was to both promote a culture of collaboration and to scale down the staff grading workload (by a factor of 1.33-1.56 – as not all worked in pairs). We promised students at the beginning of the semester *at least two rounds of iterative feedback*. It was made clear at the beginning that *an assignment is not complete until all questions being S+*, but the possibility of still having outstanding Ns after two rounds was deliberately not thoroughly discussed to avoid cognitive overload and distracting students from focusing on the first assignment.

4.2 Instructor validation of incomplete work

Upon the release of the second round feedback for the first assignment, we announced the default protocol for outstanding Not Yets.[4] Students were told to consult the instructor or a TA in office hours, resolve any confusions and obtain a working solution, then **verbally present to the instructor their updated work after class** to get their grade on the question bumped to a Satisfactory. Students were allowed to validate their work for any assignment until the end of the semester. We chose this process as the *last stage* of our feedback process for human-graded assignments due to the following considerations:

- Ideally, our assignments were calibrated to be reasonably challenging (see Section 6 for student perceptions) such that students with adequate mastery of the concepts should be able to achieve an S+ within two rounds. Any outstanding Ns after two rounds thus indicated either misconceptions or falling behind the (typical) pace of the class. By essentially *requiring* students under both situations to seek help, we were *nudging students who needed help to seek help*, essentially implementing Weber's idea [48].
- Requiring the instructor validation to be *verbal* reduced the possibility that students came to the solution under help without full understanding, since each validation served as a lightweight oral exam within the scope of one assignment question.
- Many students who would not otherwise seek one-on-one interactions with any teaching staff were *required* to do so, and we

[2]We started with a 1-hour cooldown period to prevent students from "spamming" the questions, which was quickly lowered to 15 minutes starting from the second module.

[3]This was further relaxed to groups of three for the last two modules.

[4]This technically applied also to Unassessable works, but these were naturally rare as Unassessable was usually only used when students did not turn in any work.

believed that having this experience would help the students feel more connected with the teaching staff.

- Since attendance was not tracked or tied to the students' grades in any sense, validating after class would incentivize students to attend class which we felt would be beneficial.

In sum, completion of all modules was required for almost all letter grades, and achieving S+ in all questions in the human-graded assignment was required for completing each module. This was the *Satisfactory for all* goal we pursued: ***every student* (that passes with a C- or above) *reaches an S+ in all assignment questions***.

4.3 Staff Grading Process/Management

Upon round 1 submission of each assignment, the head grader created question-specific rubrics and graded a portion of student submissions, then assigned TAs to problems based on experience and the expected time to grade. During a synchronous meeting, TAs individually worked through the grading samples and compared their results with the head grader before grading the remainder of the submissions. Synchronous meetings allowed for quick feedback and calibration on rubrics, especially if multiple TAs worked on a problem. TAs typically finished their assigned grading within 2.5 hours of synchronous meeting time. Occasionally (24/109 = 22.0% of time), they wrapped up the following day on their own time after having been calibrated. After round 1 grading finished, the head grader spot checked for consistency then released feedback to students, after which they had around one week to submit round 2.

While some alternative grading strategies prioritize reducing staff workload [29, 46], *this was not one of our motivating factors* in using ESNU. Rather, we instructed TAs to reallocate time to providing formative feedback: that all Not Yet works should either receive a self-explanatory rubric item or personalized text feedback.

Our offering had 17 TAs that graded, partitioned into one group of 8 that graded odd modules, and one group of 9 that graded even modules. Two grading groups enabled that while the current module's round 1 is graded synchronously, the previous module's round 2 could be graded asynchronously by the same TAs that graded its round 1 (so no additional calibration is needed). This gave all TAs consistent work, which can be easier to manage. TAs expressed their appreciation of this design in a staff-side check-in attempt in the 5th week, describing it as "very well-structured".

The success of our grading model hinges on a head grader with high initiative and follow through. Besides ensuring the quality of round 1 grading for the current module, the head grader must also keep tabs on round 2 grading for the previous one by consistently checking in on TAs until they finish. For a grading model as complicated as ours, we recommend clearly communicating to the head grader the required management tasks as the success of quality control can be the difference between a smooth grading experience (for both the course staff and the students) versus one where the multiple moving parts stall.

5 GRADING RESULTS

Figure 1 plots the grading results for human-graded assignments per each module/assignment. Although we did not plot the results per question due to space constraints, in all modules except the outlier module 5 (detailed in Section 5.1), 80% or more of students managed to achieve an S+ in most questions, in line with our expectation. 57.6%-84.6% of students successfully completed these assignments (compared to only 32.8% in module 5) within two rounds without having to verbally consult the instructor on any question. In particular, 29.1%-61.8% of the students successfully completed the assignments *in round 2* (the unshaded portions in Figure 1) after receiving staff feedback in round 1.

Instructor validation workload. We used the instructor validation model for all modules except modules 5 and 8 (detailed later). The number of *submissions* needing validation for the 37 questions in these modules were controllable (mean = 8.47, std = 6.23, range = $[0, 26]$), amounting to a total of 267 interactions. The overall workload would have been very manageable for the instructor *had it been spread evenly across the semester*; each interaction typically only took a few minutes, including the necessary administration to update the grades. However, as documented by the community [16, 42, 46, 48], *some students procrastinate until the very end when the deadline is the end of the semester*, and our course was no exception. The validation demand stayed low (utilized by students who did *not* procrastinate) throughout the semester, until it spiked in the last two weeks. We responded to the spiking demand of interactions with the following:

- When too many students were present, the instructor *grouped* interactions together, so that 2–5 students would have their work on the same question validated at once. We tried to get each student involved by having them take turns continuing the verbal elaboration of the proof/solution, but these experiences were undoubtedly degraded compared to one-on-one consultation.
- Students who fell *too much behind* (having 3 or more questions to validate or missed entire assignments) were told to hold off their validations and arrange separate meetings with the instructor.
- We relaxed the completion criteria for module 8 so that a completion could be achieved by getting an S+ in 6 or more of the 8 graded parts. This decision would have likely been made regardless of the demand spike, as there was insufficient time (4 days) between the round 2 feedback of the last assignment and the end of the semester for students to seek consultation in office hours.

Two strategies we did not implement but would suggest and consider for the future are (1) delegating some of the validation to very experienced staff members (such as the head grader) and (2) moving some validations to short, scheduled office hour appointments.

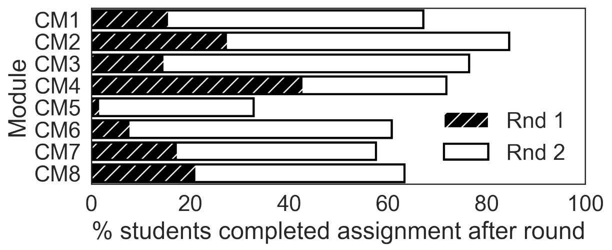
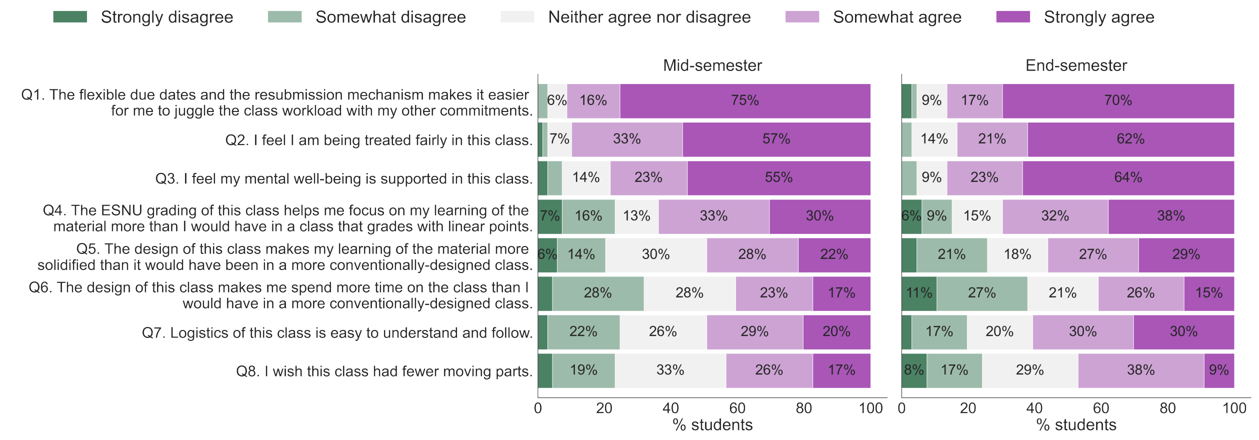
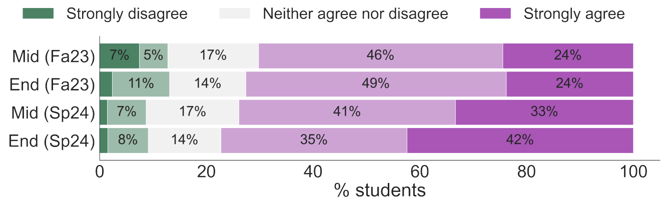


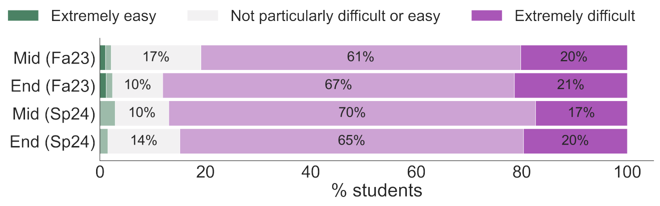
Figure 1: Grading result by module/assignment before instructor validations.



(a) Questions specific to the Spring 2024 offering (reordered and numbered for presentation).



(b) "The feedback that I receive on (human) graded assignments is valuable."



(c) "How difficult did you find the homework assignments, on average?"

Figure 2: Distribution of Likert question responses in Mid and End surveys.

5.1 Handling student struggles in writing error-free induction proofs

What is not shown in Figure 1 is that the 3 questions with the lowest S+ rate (in both round 1 and round 2) were *all 3 questions in module 5* (inductions). We believe this simply shows the (relative) difficulty in writing error-free induction proofs as novices: students could suffer from all kinds of common mistakes such as *missing or incomplete base cases*, *ill-formed inductive hypothesis*, or simply an *incomplete/incorrect induction step*, all well-documented by literature [10, 40]; these mistakes would have merely incurred a minor point penalty in traditional grading systems but rendered the entire work a Not Yet in our grading model. It appears that students need more rounds of feedback to acquire complete mastery in writing induction proofs, and they are also more prone to errors in subsequent practices *even after having acquired mastery*.

To handle the numerous Not Yets in module 5, we implemented a make-up assignment in PrairieLearn by converting each valid solution to the original assignment questions into a Proof Blocks [34] style question with distractors corresponding to all common mistakes. We required students to successfully complete *all* solutions for each question with an unresolved Not Yet in module 5 in lieu of instructor validation. This replaced more than 100 validations that would have needed to take place.

6 STUDENT PERCEPTIONS

We surveyed students' perceptions of the course policy/design and their learning at three snapshots during the semester: Pre (1st week), Mid (7th week), and End (14th week). This section is based on data from 71 students (out of 138, 51.4%) that consented to data usage. In Mid and End, students reported their sentiments on various aspects of the course via Likert scales. Figure 2a summarizes the responses to questions specific to our offering. Two additional relevant questions were also asked in the Fall 2023 offering; Figures 2b and 2c compare responses to these two questions across both semesters.

A's for all positives. Among the questions in Figure 2a, a majority of students gave a "Strongly agree" to Q1 (flexibility), Q2 (fairness), and Q3 (mental support). We view these as direct (Q1) and indirect (Q2/3) benefits from A's for all and Equitable Grading practices.

Student perceptions of this class vs. others. Q4-6 directly prompted students to compare this class to conventionally designed classes. Students (collectively) felt the ESNU scale helped them focus on learning (Q4), and their learning was more solidified under our design (Q5) without necessarily spending more time (Q6). As the semester progressed and students got more used to our design, their sentiments to Q4 and Q5 shifted towards positive, while Q6 stayed consistent.

Logistic complexity. One potential downside of adopting A's for all and Equitable Grading practices lies in the additional logistic complexity due to students' unfamiliarity and lack of clear communication from the staff [38]. However, as the semester progressed, students were able to understand and follow the policies better (Q7) and felt less overwhelmed by the complexity of course design (Q8).

Cross-offering comparison. Students perceived the feedback they received on graded assignments as more valuable than when graded using traditional points-based schemes (Figure 2b, increasingly so

Table 2: Student perceptions on course policy/design aspects. +, −, −→+, and +→− stand for *liked, disliked, disliked-then-liked,* and *liked-then-disliked,* respectively. Not all students responded to all questions. Codes with five or fewer total observations were omitted.

Topic	Pre (N = 69)		Mid (N = 71)		**Changed opinions**		End (N = 68)		**Sum**		
	+	−	+	−	−→+	+→−	+	−	+	−	**Total**
Multiple rounds for human-graded components	23	0	27	0	0	0	15	2	65	2	67
Flexible deadlines	2	2	14	1	2	1	20	3	38	7	45
Retakeable exams	17	0	1	0	0	0	3	0	21	0	21
Unlimited attempts for autograded components	11	0	1	0	0	0	1	0	13	0	13
ESNU	17	15	12	9	13	3	12	6	54	33	87
Specification grading	3	0	0	1	7	2	1	4	11	7	18
Progress tracking	1	4	0	2	0	1	0	7	1	14	15
Overlapping assignment cycles	0	1	0	5	0	0	0	3	0	9	9
Flipped learning	6	5	2	8	7	3	2	5	17	21	38
Recitations	0	0	1	2	6	2	2	0	9	4	13
Active learning in class meetings	0	0	1	3	2	4	1	2	4	9	13
Feedback received from human-grading	2	0	0	5	0	2	1	1	3	8	11
Group work in human-graded components	2	4	0	1	0	0	1	2	3	7	10

as the semester progressed) while the perceived difficulty remained stable (Figure 2c). We attribute this to the simplicity ESNU brings: the meanings of the four scale tiers are preserved universally across all questions, making it easier for staff to design and maintain the question-specific rubrics and for students to interpret the feedback.

6.1 Lessons learned from student sentiments

Prior to responding to Pre, students watched videos that outlined the course policy/design then identified one or more items that they *liked* the most and one or more that they *disliked* the most in free-form text fields. These questions were repeated in both Mid and End. In Mid, students were allowed to separately mention one item that they *initially disliked but then liked* and/or one that they *initially liked but then disliked*, if applicable. We coded students' responses iteratively based on a predetermined codebook.[5] Table 2 summarizes student sentiments on each course policy/design choice.

Stable Positives. Students appreciated various A's for all and Equitable Grading designs throughout the semester (the first four rows in Table 2). Among these, Multiple rounds for human-graded components received the most positive sentiments (65) while Flexible deadlines had the most negative sentiments (7/45). The students in the latter group pointed out that it *"let me procrastinate as much as I wanted"*, and they *"ended up doing a lot of work in the last week"*.[6] Student procrastination has been well-documented in related reports [42, 46, 48], and we note that our class was a middle ground between no flexibility and complete flexibility: both rounds of human-graded components came with deadlines while the validation and PL assignments were completely flexible. Therefore, no students really procrastinated on *everything*.

Changed opinions. In Mid, many students self-reported liking ESNU and Specification grading (rows 5-6) after disliking them initially. Students reported these designs were *"too confusing at first"*, but *"become much easier to navigate once I understand them"* and were *"fairer than I imagined it would be"*. We hypothesize such initial negativity mostly happens when students first experience Equitable Grading designs, and their impression improves with exposure.

Areas for improvement. Progress tracking and Overlapping assignment cycles (rows 7-8) received mostly negative sentiments. As we did not fully automate synchronizing student learning progress across different platforms, students were left to track their own progress and reported *"it's hard to keep up whether you have the completion status or not"*. Some students also expressed frustrations towards *"the amount of moving parts"* (also see Q6 in Figure 2a) and *"how round 2 for the previous module tended to overlap with round 1 for the current module"*. A universal "dashboard" as demonstrated in related works [1, 18] would have likely made it easier for students to keep track of all the moving parts.

Student awareness. A mild surprise throughout our experience was how much students were able to understand and acknowledge the purposes of each design, regardless of whether they liked it. This is best summarized by the following comment: *"This class has made me realize that I'm actually not that good at learning, but I'm excellent at getting good grades. I understand that this is the whole point of the grading system, but that doesn't mean I like it."*

7 CONCLUSIONS AND OUTCOME

Upon first glance, it may seem that A's for all is only possible in fully automate-able courses. This report documents our fairly successful attempt in adapting it into the DM context. Among our 138 students, two withdrew from the course; all remaining 136 mastered all core modules after the pragmatic concessions documented in Section 5.

ACKNOWLEDGMENTS

We thank all TAs and students for their participation in and contribution to this experience. We thank Jonathan Liu, Annapurna Vadaparty, Elexandra Tran, Emileaf Sun, and the anonymous reviewers for their invaluable feedback. We thank Susan Rodger and Kristin Stephens-Martinez for their mentorship and encouragement. Shao-Heng Ko was supported by the Anne T. and Robert M. Bass Endowment Fund via the Bass Instructional Fellowship.

[5] All three authors agreed upon the codebook before any coding happened. Only one code (Overlapping assignment cycles) was added to the codebook during the process. All three authors coded the *liked* and *disliked* questions in Mid independently, for which the results achieved a Fleiss' Kappa [13] $\kappa > 0.8$ after one round of calibration. The remaining parts of the data were coded independently.

[6] Interestingly, there was a student that put Flexible deadlines as both the policy that they liked the most and the policy they disliked the most.

REFERENCES

[1] Connor Robert Bernard, Dev Ahluwalia, Madelen Flores, Wilson Chu, Yuanhan Li, and Daniel D. Garcia. 2024. Supporting Mastery Learning Through an Adaptive Grade Portal. In *ACM SIGCSE TS*. 1568–1569. https://doi.org/10.1145/3626253.3635621

[2] Andrew Berns, J. Philip East, and J. Ben Schafer. 2021. Grading for Equity: A Curriculum Development and Grading Process to Enhance Instruction. In *ACM SIGCSE TS*. 1351. https://doi.org/10.1145/3408877.3432503

[3] Benjamin S Bloom. 1968. Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment* 1, 2 (1968), n2. https://eric.ed.gov/?id=ED053419

[4] Jennifer Campbell, Andrew Petersen, and Jacqueline Smith. 2019. Self-paced Mastery Learning CS1. In *ACM SIGCSE TS*. 955–961. https://doi.org/10.1145/3287324.3287481

[5] Lijun Chen, Joshua A. Grochow, Ryan Layer, and Michael Levet. 2022. Experience Report: Standards-Based Grading at Scale in Algorithms. In *ACM ITiCSE*. 221–227. https://doi.org/10.1145/3502718.3524750

[6] Andrew A Cooper. 2020. Techniques grading: Mastery grading for proofs courses. *Primus* 30, 8-10 (2020), 1071–1086. https://doi.org/10.1080/10511970.2020.1733151

[7] Catherine H. Crouch and Eric Mazur. 2001. Peer Instruction: Ten years of experience and results. *American Journal of Physics* 69, 9 (2001), 970–977. https://doi.org/10.1119/1.1374249

[8] Adrienne Decker, Stephen H. Edwards, Brian M. McSkimming, Bob Edmison, Audrey Rorrer, and Manuel A. Pérez-Quiñones. 2024. Transforming Grading Practices in the Computing Education Community. In *ACM SIGCSE TS*. 276–282. https://doi.org/10.1145/3626252.3630953

[9] Kezia Devathasan, Jason Kepler, Johnathan Warawa, Amy Penney, Isabella Tsui, and Celina Berg. 2023. PrairieLearn in CS1: An Experience Report. In *ACM WCCCE*. 10:1–10:2. https://doi.org/10.1145/3593342.3593344

[10] Robert L. Scot Drysdale. 2011. Mathematical induction is a recursive technique. In *ACM SIGCSE TS*. 269–274. https://doi.org/10.1145/1953163.1953246

[11] Duke CompSci230 SP24 Course Website. 2024. https://sites.duke.edu/compsci230sp2024/.

[12] Joe Feldman. 2018. *Grading for equity: What it is, why it matters, and how it can transform schools and classrooms.*

[13] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378. https://doi.org/10.1037/h0031619

[14] Dan Garcia, Maria Camarena, Kevin Lin, and Jill Westerlund. 2023. Equitable Grading Best Practices. In *ACM SIGCSE TS*. 1200–1201. https://doi.org/10.1145/3545947.3569602

[15] Dan Garcia, Armando Fox, Solomon Russell, Edwin Ambrosio, Neal Terrell, Mariana Silva, Matthew West, Craig B. Zilles, and Fuzail Shakir. 2023. A's for All (As Time and Interest Allow). In *ACM SIGCSE TS*. 1042–1048. https://doi.org/10.1145/3545945.3569847

[16] Dan Garcia, Jim Huggins, Lauren J. Bricker, Adam Gaweda, David J. Malan, Joël Porquet-Lupine, and Kristin Stephens-Martinez. 2023. It Seemed Like a Good Idea at the Time: ("Let Me Help You with That" edition). In *ACM SIGCSE TS*. 1204–1205. https://doi.org/10.1145/3545947.3569599

[17] Dan Garcia, Connor McMahon, Yuan Garcia, Matthew West, and Craig B. Zilles. 2022. Achieving "A's for All (as Time and Interest Allow)". In *ACM L@S*. 255–258. https://doi.org/10.1145/3491140.3528289

[18] Dan Garcia, Connor McMahon, Yuan Garcia, Matthew West, and Craig B. Zilles. 2022. Software Support for "A's for All". In *ACM L@S*. 475–476. https://doi.org/10.1145/3491140.3528260

[19] Dan Garcia, Connor McMahon, Yuan Garcia, Craig B. Zilles, Matthew West, Mariana Silva, Solomon Russell, Edwin Ambrosio, and Neal Terrell. 2023. Actually Achieving "A's for All" (As Time and Interest Allow). In *ACM SIGCSE TS*. 1175. https://doi.org/10.1145/3545947.3569633

[20] Gradescope. 2024. https://www.gradescope.com/.

[21] Georgiana Haldeman, Monica Babes-Vroman, Andrew Tjang, and Thu D. Nguyen. 2021. CSF: Formative Feedback in Autograding. *ACM Trans. Comput. Educ.* 21, 3 (2021), 21:1–21:30. https://doi.org/10.1145/3445983

[22] Qiang Hao, David H. Smith IV, Lu Ding, Amy J. Ko, Camille Ottaway, Jack Wilson, Kai Arakawa, Alistair Turcan, Timothy Poehlman, and Tyler Greer. 2022. Towards understanding the effective design of automated formative feedback for programming assignments. *Comput. Sci. Educ.* 32, 1 (2022), 105–127. https://doi.org/10.1080/08993408.2020.1860408

[23] Brian Harrington, Abdalaziz Galal, Rohita Nalluri, Faiza Nasiha, and Anagha Vadarevu. 2024. Specifications and Contract Grading in Computer Science Education. In *ACM SIGCSE TS*. 477–483. https://doi.org/10.1145/3626252.3630929

[24] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2019. A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Trans. Comput. Educ.* 19, 1 (2019), 3:1–3:43. https://doi.org/10.1145/3231711

[25] David L. Largent, Christian Roberson, Linda F. Wilson, and Manuel A. Pérez-Quiñones. 2024. Transform Your Computer Science Course with Specifications Grading. In *ACM SIGCSE TS*. https://doi.org/10.1145/3626253.3633426, 1900.

[26] Kevin Lin. 2022. Specifications Grading Policies. https://kevinl.info/specifications-grading-policies/

[27] Kevin Lin. 2024. Alternative Grading. https://kevinl.info/alternative-grading/

[28] Firas Moosvi, Dirk Eddelbuettel, Craig B. Zilles, Steven A. Wolfman, Fraida Fund, Laura K. Alford, and Jonatan Schroeder. 2023. Creating Algorithmically Generated Questions Using a Modern, Open-sourced, Online Platform: PrairieLearn. In *ACM SIGCSE TS*. 1177. https://doi.org/10.1145/3545947.3569634

[29] Linda B Nilson and Claudia J Stanny. 2015. *Specifications grading: Restoring rigor, motivating students, and saving faculty time.*

[30] Claudia Ott, Brendan McCane, and Nick Meek. 2021. Mastery Learning in CS1 - An Invitation to Procrastinate?: Reflecting on Six Years of Mastery Learning. In *ACM ITiCSE*. 18–24. https://doi.org/10.1145/3430665.3456321

[31] Leah Perlmutter, Jayne Everson, Ken Yasuhara, Brett Wortzman, and Kevin Lin. 2022. Reading Between the Lines: Student Experiences of Resubmission in an Introductory CS Course. In *ACM SIGCSE TS*. 1137. https://doi.org/10.1145/3478432.3499112

[32] Seth Poulsen, Sami Sarsa, James Prather, Juho Leinonen, Brett A. Becker, Arto Hellas, Paul Denny, and Brent N. Reeves. 2024. Solving Proof Block Problems Using Large Language Models. In *ACM SIGCSE TS*. 1063–1069. https://doi.org/10.1145/3626252.3630928

[33] Seth Poulsen, Mahesh Viswanathan, Geoffrey L. Herman, and Matthew West. 2021. Evaluating Proof Blocks Problems as Exam Questions. In *ACM ICER*. 157–168. https://doi.org/10.1145/3446871.3469741

[34] Seth Poulsen, Mahesh Viswanathan, Geoffrey L. Herman, and Matthew West. 2022. Proof Blocks: Autogradable Scaffolding Activities for Learning to Write Proofs. In *ACM ITiCSE*. 428–434. https://doi.org/10.1145/3502718.3524774

[35] William J. Rapaport. 2011. A Triage Theory of Grading: The Good, the Bad, and the Middling. *Teaching Philosophy* 34, 4 (2011), 347–372. https://doi.org/10.5840/teachphil201134447

[36] Jordan Schwartz, Madison Bohannan, Jacob Yim, Yuerou Tang, Dana Benedicto, Charisse Liu, Armando Fox, Lisa Yan, and Narges Norouzi. 2024. Automated Support for Flexible Extensions. In *ACM SIGCSE TS*. 1810–1811. https://doi.org/10.1145/3626253.3635628

[37] Aayush Shah, Alan Lee, Chris Chi, Ruiwei Xiao, Pranav Sukumar, Jesus Villalobos, and Dan Garcia. 2022. Improved Testing of PrairieLearn Question Generators. In *ACM SIGCSE TS*. 1165. https://doi.org/10.1145/3478432.3499113

[38] Eman Sherif, Jayne Everson, F. Megumi Kivuva, Mara Kirdani-Ryan, and Amy J. Ko. 2024. Exploring the Impact of Assessment Policies on Marginalized Students' Experiences in Post-Secondary Programming Courses. In *ACM ICER*. 233–245. https://doi.org/10.1145/3632620.3671100

[39] Michael Shindler, Matt Ferland, Aaron Cote, and Olivera Grujic. 2020. Experience report: preemptive final exams for computer science theory classes. *J. Comput. Sci. Coll.* 35, 10 (2020), 9–14. https://dl.acm.org/doi/10.5555/3417699.3417700

[40] Thérèse Smith and Robert McCartney. 2014. Computer science students' concepts of proof by induction. In *ACM Koli Calling*. 51–60. https://doi.org/10.1145/2674683.2674696

[41] Katherine E Stange. 2018. Standards-based grading in an introduction to abstract mathematics course. *Primus* 28, 9 (2018), 797–820. https://doi.org/10.1080/10511970.2017.1408044

[42] Kristin Stephens-Martinez. 2022. Soft, hard, and late deadlines (or another failed experiment). https://ksm-cs.blogspot.com/2022/02/soft-hard-and-late-deadlines-or-another.html

[43] Robert Talbert. 2022. Grading in my Discrete Mathematics class: a 3x3x3 reflection. https://rtalbert.org/grading-in-my-discrete-mathematics-class-a-3x3x3-reflection/

[44] Yuerou Tang, Jacob Yim, Jordan Schwartz, Madison Bohannan, Dana Benedicto, Charisse Liu, Armando Fox, Lisa Yan, and Narges Norouzi. 2024. Supporting Mastery Learning with Flexible Extensions. In *ACM SIGCSE TS*. 1834–1835. https://doi.org/10.1145/3626253.3635615

[45] Cynthia Bagier Taylor, Jaime Spacco, David P. Bunde, Andrew Petersen, Soohyun Nam Liao, and Leo Porter. 2018. A multi-institution exploration of peer instruction in practice. In *ACM ITiCSE*. 308–313. https://doi.org/10.1145/3197091.3197144

[46] Ella Tuson and Tim Hickey. 2022. Mastery Learning and Specs Grading in Discrete Math. In *ACM ITiCSE*. 19–25. https://doi.org/10.1145/3502718.3524766

[47] Ella Tuson and Timothy J. Hickey. 2023. Mastery Learning with Specs Grading for Programming Courses. In *ACM SIGCSE TS*. 1049–1054. https://doi.org/10.1145/3545945.3569853

[48] Robbie Weber. 2023. Using Alternative Grading in a Non-Major Algorithms Course. In *ACM SIGCSE TS*. 638–644. https://doi.org/10.1145/3545945.3569765

[49] Matthew West, Geoffrey L Herman, and Craig Zilles. 2015. Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In *ASEE Annual Conference & Exposition*. 26–1238. https://doi.org/10.18260/p.24575

[50] Chenyan Zhao, Mariana Silva, and Seth Poulsen. 2024. Autograding Mathematical Induction Proofs with Natural Language Processing. arXiv:2406.10268 [cs.AI] https://arxiv.org/abs/2406.10268