A Flexible Formative/Summative Grading System for Large Courses

Albert Lionelle
Northeastern University
a.lionelle@northeastern.edu

Sudipto Ghosh
Colorado State University
ghosh@cs.colostate.edu

Marcia Moraes
Colorado State University
Marcia.Moraes@colostate.edu

Tran Winick
Colorado State University
tran.lieu@colostate.edu

Lindsey Nielsen
Colorado State University
lindsey.rebecca.nielsen@colostate.edu

ABSTRACT

Students in entry level CS courses come from diverse backgrounds and are learning study and time management skills. Our belief for their success is that they must master a growth mindset and that the final grade should represent their final mastery of topics in the course. Traditional grading systems tend to be too restrictive and hinder a growth mindset. They require strict deadlines that fail to easily account for student accommodations and learning differences. Furthermore, they run into averaging and scaling issues with 59% of a score counting as failing, making it difficult for students to redeem grades even if they later demonstrate mastery of topics.

We designed a formative/summative grading system in our CS0 and CS1 classes for both on-campus and online students to support a structured growth mindset. Students can redo formative assignments and are provided flexible deadlines. They demonstrate their mastery in summative assignments. While being inspired by other grading systems, our system works seamlessly with auto-grading tools used in large, structured courses. Despite the flexibility, the courses provided a level of rigor before allowing students to continue onto the next course.

Overall, 65% of students resubmitted assignments increasing their scores, participated in ungraded assignments, and used formative assignments for additional practice without a distinction between race or gender. These students went to the traditional follow-on CS2 course and 94% passed compared with 71% who took CS1 with a traditional grading system.

CCS CONCEPTS

• **Social and professional topics** → *Computational thinking*; *Computer science education.*

KEYWORDS

Computing education, CS 0, CS1, grading, growth mindset

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569810

ACM Reference Format:
Albert Lionelle, Sudipto Ghosh, Marcia Moraes, Tran Winick, and Lindsey Nielsen. 2023. A Flexible Formative/Summative Grading System for Large Courses. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569810

1 INTRODUCTION

The grading system in a course should support our pedagogy. If we believe that students utilizing a growth mindset towards learning and computer science can develop mastery [6, 11, 16, 18, 26, 38], we must ask if the grading system supports learning from mistakes, redoing assignments, and overcoming challenges. Furthermore, if we want to support students from diverse backgrounds, does the grading system support their diverse needs and accommodations?

A traditional grading system is typically based on a 100-point scale, with 59% considered failing, and the remaining 41% passing at various ranking levels. Students' ranking in these systems is based on the culmination of assignments at snapshots in time utilizing static due dates. Thus, students are expected to follow a singular, specific path in which every assignment builds on the next, and their mastery is mostly consistent throughout the whole semester. While there are various ways teachers have adapted such systems, it has been argued that the system is inherently biased and inequitable [7]. It assumes all students come into the course with the same background, ability and time to work on the course, and the resources to gain mastery within the specified time ranges. Furthermore, due to the snapshot approach it takes towards calculating grades, traditional grading is counter-intuitive to the idea that students can grow and master topics later. This is compounded by the averaging and scaling factor that happens when failing grades account for 59% of all possible scores, thus requiring nine assignments at 100% to repair a zero grade on one assignment. Dropping grades doesn't encourage mastery unless there is a clear mechanism that separates learning and demonstrating mastery.

In order to improve retention and performance for students who come from a variety of backgrounds, we sought to modify our traditional grading system to better represent final mastery of course topics making us rethink how assignments should be handled. Students would need the ability to resubmit assignments and even exams, to perceive them as challenges to be overcome even if they have an initial setback. The grading system should be flexible to allow students to repair any previous damage to their grades. Students should be allowed to master different topics at varied rates within the time constraints of a semester. The system should work seamlessly both with our teaching tools, such as interactive text books, LMS quizzes, and auto graded labs, which are often based on the traditional grading system's calculation of grades, and within the environment of a traditional system requiring that we provide the standard expected letter grade.

Alternatives, such as grading for equity [7], mastery [8], contract [17], and specifications [27, 29] exist. They attempt to make grading better reflect student skill level by the end of the course. However, these systems rely on standards tracking, which we couldn't configure in our auto grader and interactive textbook, or on feedback loops between teacher and student, which do not scale for large courses. These systems inspired us to develop a Formative/Summative grading system for use in CS0 and CS1 to provide students the opportunity to learn, grow, and flexibly show mastery of the topics while maintaining course rigor.

Formative assignments are those that students could redo as many times as they wanted, turn in again, even late, without penalty. We gave students full flexibility to use them to learn the content. It also takes into account that students come from different backgrounds and some may need more time on topics. Others may need to move deadlines to reflect life commitments without always having to ask for instructor permission. Formative assignments made up at least 60% of the final grade to compensate for the traditional grade scaling factors. We had a small number of Summative assignments, which had a limited number of submission attempts and strict deadlines. They helped students demonstrate what they learned in different ways. All assignments were scaled to between 0-4, mapping each point to common growth mindset terms to encourage students towards completion. We applied this grading system across CS0 and CS1, to both on-campus and online courses, in both Fall and Spring semesters. We answered the following questions:

(1) Do students catch up? Or do they at least use the flexibility in both submissions and late windows to increase their grade after the traditional due dates?
(2) Do students from underrepresented populations utilize the retake attempts and catch up opportunities differently?
(3) Do students continue to perform when these supports are removed (e.g., their later CS2 course that uses a traditional grading system)?

Overall, our goal was to develop a grading system that was flexible, student and learning focused, and maintained course rigor, student performance, and retention. Developing the system challenges the assumptions we often make about essential elements of a course, like strict deadlines. The following experience report shows that students not only made heavy use of the system in CS0 and CS1, but also performed better in a traditionally graded CS2.

2 RELATED WORKS

Traditional grading has shaped how Computer Science (CS) has been taught for many decades, and it has been challenged within CS [2, 28] and outside [12, 25, 31]. This section summarizes alternative grading research that influenced the design of our approach.

Feldman [7] argues that grading for equity instead of traditional grading can aid a student's learning process by considering three principles: accuracy, bias-resistance, and motivation. Grading for equity can be implemented using standards focused approaches, clear rubrics, 0-4 point grading scale, mode-based grading, encouragement for resubmissions, no late penalties, and no extra credit.

Specifications grading treats grading as binary. Partial credit is replaced by feedback and the ability to resubmit assignments until success [23]. CS courses using this system showed higher scores, and positive student feedback [27]. Sanft et al. [29] determined that the time spent grading programs was less than that of traditional partial grading, but more time was needed to provide feedback.

Nee and Ramirez [22] and Ukpokodu [35] found that allowing resubmissions with timely feedback not only encouraged assignment completion and increased the desire to learn the content, but also enhanced the sense of belonging within the course and the students' sense of support from the department. Barker et al. [1] and Sax et al. [30] found that feeling supported by the department, professors, and peers is a significant indicator of the sense of belonging of underrepresented students in computing.

Kuhn [14] states that merit grading has contributed to our current equity gaps and recommends replacing it with contract grading, where each student and instructor agree on a set of assignments that the student needs to finish in order to achieve a desired course grade. There is no penalty for late assignments.

While Svartdal et al. [34] argue that removing late penalties leads to procrastination, Martin et al. [19] provides three techniques to reduce procrastination: create schedules, reflective writing, and directed email reminders. Of these three, directed email reminders was the only one demonstrated to reduce procrastination.

Mastery grading focuses on stops in progression until they show mastery of concepts. Kulik et al. [15] and Garner et al. [8] present various ways mastery grading has been used and how it positively influenced student learning in areas such as math, science, social sciences and CS. Mastery grading has been used to let students progress at their own pace with general positive feedback from students [5, 20], improvement in course success rates [32], and showing that students who used mastery grading had more success in future CS courses in comparison to students who were graded more traditionally [20]. Lejeune [17] combined contract grading with mastery grading as a means to better show mastery requirements to students.

O'Malley and Aggarwal [24] and Khanna [13] suggest that ungraded quizzes done as active retrieval of content contributed toward students' better performance on final exams. These findings are in accordance with previous research on the testing effect on students' performance [3, 4, 10, 36].

All the above alternative grading systems share a common characteristic: they encourage students to keep working on assignments until they have met that learning goal. This is the major characteristic of a growth mindset, which, in a national review of K-12 education, has been shown to promote learning and retention, especially among lower achieving students [38]. A student may not understand a concept immediately, but providing a structure in which they can continue to work on something until they learn it promotes this mindset. In CS, a growth mindset is integral to student success [16, 26, 33], but Murphy and Thomas [21] state that this is a difficult topic to teach. Even if faculty express a growth mindset in their teaching practices [9], intentional pedagogy and support must be built into the curriculum to promote the same mindset in students [18].

3 DESIGN

Our grading system groups assignments into formative and summative categories. Formative assignments are open all semester for students to redo until they get it correct. Different assignment types may have different conditions before a redo/retake attempt is allowed. Most had a recommended due date, and then a closing date at the end of the semester after which submissions would no longer be accepted. Formative assignments included weekly labs, weekly quizzes (termed knowledge checks), and some exams. Summative assignments have a limited number of attempts and often a strict deadline and included final projects, final exams, and reflections.

Students were informed that formative assignments are meant to facilitate learning and that we won't punish them for getting something wrong while learning. We wanted them to go back until they could get it right. Formative assignments would need to make up at least 60% of the grade to compensate for the averaging factor in traditional grading. Summative assignments were opportunities for them to demonstrate what they have learned and are measures of their skill level in the subject. By making it at least 40% of the grade, summative assignments could set a student's level of mastery between D-A grades. We found additional constraints were helpful in motivating students to do the work and make adequate progress in the course.

Growth Focused Rubric: Assignments were shifted to a 4 point scale with the grades being growth mindset focused wording of "Learning", "Approaching", "Meets", and "Exceeds". This rubric was the basis of auto-graded assignments, including setting up our coding unit tests as increasing tiers of difficultly to achieve reporting feedback using those terms.

Mastery of Knowledge Checks / Weekly Quizzes: Weekly quizzes were given with random questions pulled from question banks. If they didn't finish a weekly knowledge check with a 3 or 4, they wouldn't be able to open up the next week's readings, labs, lecture slides, and assignments. To reduce procrastination, students received emails if they missed the weekly deadline.

Formative Exams: We wanted exams to be opportunities for students to gauge themselves in a proctored environment. To make sure students were studying between exam attempts, we required the students to go back and earn 100% on readings, 100% on knowledge checks, and 80% on the labs in the unit.

Tracking Ungraded Assignments: In CS1 we had ungraded interactive reading assignments. Even though they were ungraded, students saw tracked points with recommended due dates, but knew the scores would not affect their grade.

Liberal Drop Policy: We found it beneficial to be liberal in our policy of dropping the lowest grade for the formative assignments. We believed students don't need every formative to meet the same skill level as other students, and this gave them the opportunity to choose formative assignments.

3.1 Course Setup

Between Fall 2021 and Spring 2022, both CS0 and CS1 courses were set up using formative and summative grading. Due to the differences in the courses, there are variations in the actual distribution of grades and how different categories were handled.

3.1.1 *CS0: Python.*

Formative assignments included interactive readings and coding labs, which were autograded and could be resubmitted as many times as they wanted, along with knowledge checks. They had 3 of 4 essays that were formative, and would only be resubmitted after they spoke with a TA. Summative assignments were the midterm exam, the final exam, and a final practical project that involved both a coding and written portion. A 70/30 split was used for the formative and summative.

3.1.2 *CS1: Java.*

Like many universities, CS1 is broken up into prior and no-prior programming experience groups to reduce the intimidation factor [37]. They do the same assignments, and a 60/40 split was used for the formative and summative. In addition to knowledge checks and labs, CS1 had 3 of 4 exams formative, and an ungraded interactive reading category. Exams were generated from question banks, making every exam slightly different. For the summative category, they had a final exam, multiple coding exams, reflective writing, and a final multi-week project.

3.2 Research Setup

To evaluate the benefits of the grading system, we tracked both student scores when the assignment was due, and further submission attempts throughout the course. Students were tracked both Fall and Spring, and the Fall CS1 cohort was also tracked in Spring CS2 that utilized a more traditional model to see performance compared to students who learned CS1 by the traditional grading system.

4 RESULTS

Measuring the difference between due dates and the final grade provides a sense of how many students went back to increase their formative grades until mastery. We found higher formative grades also led to higher summative grades, median grade increase was 17 points across all 525 students who opted into the study for both semesters. Shown in table 1, 65% of students increased 10 points or more, a letter grade, by going back and redoing assignments after the due date. The increase was significant for all sections.

Table 1: Overall grade change from due date to final grade.

Group	Median Score Change	Percent Who Increased a Letter Grade
All Courses	17 ($p < 0.01$)	65%
CS0: Fall 21	26 ($p < 0.01$)	81%
CS0: Spring 22	14 ($p < 0.01$)	60%
CS0: All	20 ($p < 0.01$)	69%
CS1: Fall 21	11 ($p < 0.01$)	52%
CS1: Spring 22	17 ($p < 0.01$)	69%
CS1: All	15 ($p < 0.01$)	60%

When grouped by final grade, 54% of A students had increased a letter grade, i.e., 46% of A students often had A grades by the due date. B students show 77% increased at least a letter grade from C or lower to a B before the end of the course. Table 2 shows the median increase, along with significant difference based on various group cross sections of the course.

There wasn't a significant difference between male and female identifying students as seen in Figure 1. The median grade of both

Table 2: Grade change across all courses, all semesters.

Group	Median Score Change	Percent Who Increased a Letter Grade
Final Letter Grade (ANOVA: p <0.01)		
A	12 (p <0.01)	54%
B	23 (p <0.01)	77%
C	28 (p <0.01)	78%
D	27 (p <0.01)	90%
F	10 (p <0.01)	n/a
Gender (ANOVA: p = 0.6)		
Male	17 (p <0.01)	64%
Female	16 (p <0.01)	66%
Self Identified Race (ANOVA: p <0.01)		
White	17 (p <0.01)	64%
Hispanic/Latinx	18 (p <0.01)	65%
Asian	11 (p <0.01)	51%
Black	27 (p <0.01)	78%
Native American	20 (p <0.01)	69%

groups are within a point of each other, while 64% and 66% increased their grade by 10 points or more.

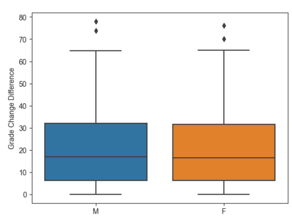

Figure 1: Grade change grouped by gender.

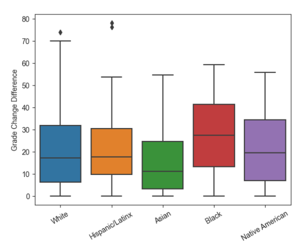

Figure 2: Grade change grouped by race.

Black students increased the highest of any group with a median increase of 27, and 78% of Black students increased a letter grade. Asian students had the lowest increase of any group with only an 11 point and 51% increase of a letter grade throughout the semester. Hawaiian/Pacific Islander numbers were too small for accurate measure. Figure 2 shows the grade differences across all courses grouped by underrepresented minority group and compared to the majority of students. An ANOVA test between the groups showed a p value of 0.002, significant at $p < 0.05$.

While the data shows students are able to increase their grades and catch up, or at least work at their own pace, a comparison to historical retention rates by course end showed minimal changes.

4.1 CS0 Results

CS0 students showed a higher movement rate across the semester, with some groups increasing their grade by 34 points (three letter grades!) on average.

Table 3 shows the breakdown of grade score differences from due date to the final grade grouped by final grade, and then grouped by semester plus delivery type. Online students had a 12 point higher increase than on-campus students. This could be attributed to the necessity of online students being able to flex due dates based on their life schedules.

Table 3: Difference between due date and final scores in CS0.

Group	Median Score Change	Percent Who Increased a Letter Grade
Final Letter Grade (ANOVA: p <0.01)		
A	13 (p <0.01)	60%
B	25 (p <0.01)	81%
C	25 (p <0.01)	75%
D	28 (p <0.01)	94%
F	12 (p <0.01)	n/a
Delivery Type and Term (ANOVA: p <0.01)		
CS 0 Fall - On Campus	26 (p <0.01)	80%
CS 0 Fall - Online	34 (p <0.01)	100%
CS 0 Spring - On Campus	13 (p <0.01)	59%
CS 0 Spring - Online	25 (p <0.01)	80%

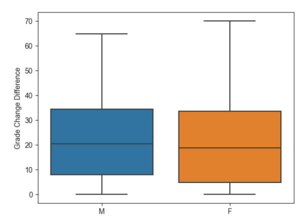

Figure 3: CS0 grade change grouped by gender.

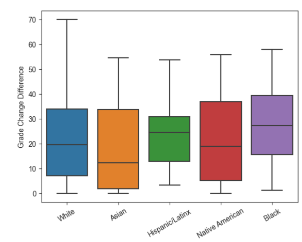

Figure 4: CS0 grade change grouped by race.

Figure 3 shows a gender comparison for students in CS0. There are very little differences between male and female identifying students. Figure 4 shows the breakdown based on students who come from underrepresented backgrounds. While Black students had the highest increase in grade at 27 points, and Asian students had the lowest at 12 points, an ANOVA test between the groups shows p = 0.19. There isn't a significant difference at $p < 0.05$.

Table 4 shows the final summative averages against formative averages. CS0 students scored noticeably higher on the formative side than the summative side, though the formative average was still predictive of a higher summative average. It is worth noting the 70/30 split meant students could have lower summative grades, and still end up with a higher grade in the course. At our university, a B or higher in CS0 is required to take CS1, which meant students going on passed both portions of their grade.

Overall, CS0 students took advantage of the flexible due dates and re-submission opportunities, often increasing more than a letter grade from the original due dates.

4.2 CS1 Results

Table 5 shows the breakdown across letter grade and section. 49% of A students increased their grade to an A, and with a median increase of 10 points meaning that in many cases they were B students pushing themselves up a letter grade. Fall semester, on-campus prior programming experience students had a median increase of

Table 4: CS 0: Formative and Summative averages.

Final Grade Group	Formative Average	Summative Average
A	97%	88%
B	89%	70%
C	84%	52%
D	72%	40%
F	36%	13%

only 3 points, and only 23% resubmitted to increase their grade. The section is mostly made up of students who had AP credit for high school programming or some other recent programming background. Contrary to this, 83% and 91% of the prior programming online students took advantage of being able to resubmit. This group of students is made up mostly of second bachelor students who have some self-taught programming and were working full time. No-prior programming experience students consistently took advantage of being able to resubmit with 61% for both semesters on-campus and 80% and 81% online.

Table 5: Breakdown of CS 1 by grade and section. Students self selected on prior programming experience.

Group	Median Score Change	Percent Who Increased a Letter Grade
Final Letter Grade (ANOVA: p <0.01)		
A	10 (p <0.01)	49%
B	22 (p <0.01)	73%
C	30 (p <0.01)	80%
D	25 (p <0.01)	88%
F	9 (p <0.01)	n/a
Delivery Type and Term (ANOVA: p <0.01)		
CS 1:No-prior Fall - On Campus	14 (p <0.01)	61%
CS 1: No-prior Fall - Online	29 (p <0.01)	80%
CS 1: Prior Fall - On Campus	3 (p <0.01)	23%
CS 1: Prior Fall - Online	37 (p <0.01)	83%
CS 1: No-prior Spring - On Campus	16 (p <0.01)	61%
CS 1: No-prior Spring - Online	40 (p <0.01)	81%
CS 1: Prior Spring - On Campus	17 (p <0.01)	74%
CS 1: Prior Spring - Online	38 (p <0.01)	91%

Figure 5 shows no noticeable gender differences for CS1 students. Figure 6 shows the differences for students from underrepresented backgrounds. Black students took the most advantage of the re-submission attempts increasing scores by 29 points and 72% increasing a letter grade or more. An ANOVA test showed p = 0.02, significant at $p < 0.05$.

Table 6 shows the formative average against the summative average, grouped by the final letter grade. The alignment between the two score categories is noticeable with both A and B students earning A and Bs on both formative and summative categories.

4.3 Ungraded Assignment Comparison

CS1 students were encouraged to do the interactive readings before lecture, which would often start with a question from the readings. The module layout displayed readings as a task to do even though the reading category did not count towards the final grade.

Figure 7 shows the Fall 2021 reading score at the due date, and the final reading score when the course was completed. Figure 8

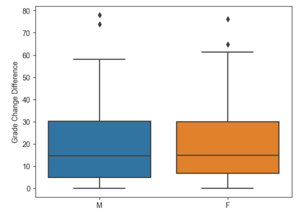

Figure 5: CS1 grade change grouped by gender.

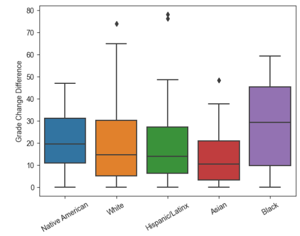

Figure 6: CS1 grade change grouped by race.

Table 6: CS 1: Formative and Summative Grade Averages

Final Grade Group	Formative Average	Summative Average
A	97%	95%
B	86%	82%
C	79%	69%
D	67%	54%
F	39%	14%

shows the Spring 2022 comparison. For both semesters, we see that anyone who earned an A, B, or C in the class would go back to the reading until they earned a 4 of 4 points on it, even though students were aware the points didn't count towards their grade.

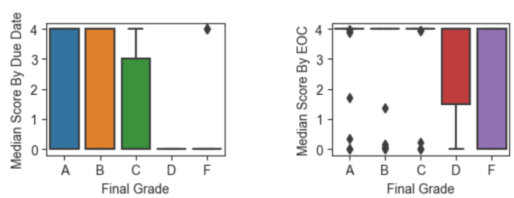

Figure 7: Fall: Reading grade score changes.

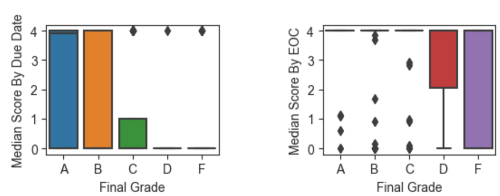

Figure 8: Spring: Reading grade score changes.

4.4 CS2 Performance

CS2: Data Structures follows a traditional grading system with strict due dates and late penalties. The one exception is weekly quizzes that can be redone without penalty for studying. The students who took CS1 in Fall 21 who learned using the formative / summative grading scheme, took CS2 in Spring 22 with students who had other entry points, such as CS1 with traditional grading schemes, community college, or the AP CS exam.

Looking at the final grade, 94% of those who took CS1 using the formative / summative grading scheme passed with a C or higher while only 71% of the other students passed with a C or higher. A 70-75% C or higher rate aligns with historical institutional averages. Figure 9 shows the comparison across grades. Comparing the overall grade counts using chi-squares a result of p = 0.0001 indicates significance at $p < 0.05$.

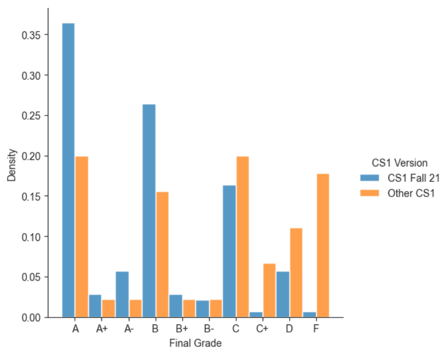

Figure 9: CS2 grade comparison based on CS1 version.

5 THREATS TO VALIDITY

The underlying philosophy of our grading system was to allow for flexibility, which meant student procrastination could occur. An internal threat to validity is the potential that students are able to complete their work in spite of procrastination with the appearance of student growth. This issue will be investigated in further research. However, the fact that students continued to perform better than their counterparts in the following class with more traditional grading systems leads us to believe that their overall study habits were not hurt, but rather, improved.

6 DISCUSSION

Our experience with building a grading scheme supportive of our pedagogy using the tools we have become accustomed to was a rewarding experience.

***Q1**. Do students catch up? Or do they at least use the flexibility in both submissions and late windows to increase their grade after the traditional due dates?* Yes. Students increased their grade from the initial due date to the subsequent grade. Resubmission attempts were especially prevalent for knowledge checks throughout the semester, wherein 20 submission attempts were common. More surprising, most students went back and worked on the reading assignments although they didn't count for a grade.

Online students used the flexibility the most. It appears this grading system is particularly well-suited for online environments, as many online students have full-time jobs requiring greater flexibility. Overall, allowing late submissions without penalty didn't hurt students, and naturally supports student needs.

***Q2**. Do students from underrepresented populations utilize the retake attempts and catch up opportunities differently?* Mixed. We found that gender did not have an influence on retakes and grade increases; and while there was a difference in racial demographics, especially for Black and Asian students, the results were not conclusive.

***Q3**. Do students continue to perform when these supports are removed such as in their later CS2 course that uses a traditional grading system?* Yes. Our goal was to ensure that the new grading system didn't hurt performance in later courses that may use traditional systems. However, the CS2 results were astounding with 94% of the students passing a course that usually has around a 75-80% C or above pass rate. Students are retaining more information and learning how to work until mastery while still meeting strict due dates in CS2.

We found the 60/40 split in CS1 for assignments to be better than the 70/30 split in CS0. Ensuring every assignment was on a 0-4 scale with a growth focused rubric was extremely helpful as it encouraged students to view assignments as ways to master topics.

The system allowed instructors to focus more on teaching than point bartering. If a student asked for extra time or credit, they were told it was already built into the grading system. At our university, accommodations are often extra time on assignments, and the system met most accommodations without individual modifications. It also allowed instructors to explain the growth mindset in relation to the grading system, how the formative assignments were to learn the content, and the summative assignments were their chance to demonstrate what they learned.

7 CONCLUSION

The goal was to create a flexible grading system that supports a growth mindset and allows diverse students with different backgrounds and needs to learn the material then accurately demonstrate what they know. Furthermore, this grading system had to work within the tools we already used such as autograders, interactive text books, and canvas style quizzes.

Students used the flexibility and resubmitted most assignments to increase their grade. Across CS0 and CS1, online and on-campus spanning two semesters, we found that 65% of the students increased their grades from initial submission to final score with a median of 17 points. Furthermore, we found CS1 had a better balance between formative and summative grading with nearly identical mapping between formative grades and summative grades by the end of the semester. We also found students who earned C or higher grades would often go back and complete ungraded assignments in order to better learn and understand the material.

Even more, we found that 94% of the students who learned using the formative/summative grading system in CS1 passed CS2 with a C or above, as compared to 71% for those who had traditional grading systems in CS1. As CS2 used a very traditional grading system, this illustrates that the Formative/Summative grading system does not hurt their study habits for future courses, and arguably, may improve their overall ability in future courses.

While this case study still leaves a number of questions, we hope to answer them by exploring this system with improved monitoring of student habits, applying it in later courses, and moving more assignments to the ungraded category while still achieving the desired student performance.

ACKNOWLEDGMENTS

This work was supported in part by funding from NSF under Award Number OAC 1931363, and conducted at Colorado State University. IRB Approval Protocol Number: 2630

REFERENCES

[1] Lecia Barker, Christopher Lynnly Hovey, and Leisa D. Thompson. Results of a large-scale, multi-institutional study of undergraduate retention in computing. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–8, 2014.

[2] Andrew Berns. Scored out of 10: Experiences with binary grading across the curriculum. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 1152–1157, 2020.

[3] Elizabeth L Bjork, Robert A Bjork, et al. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the real world: Essays illustrating fundamental contributions to society*, 2(59-68), 2011.

[4] Elizabeth Ligon Bjork, Jeri L Little, and Benjamin C Storm. Multiple-choice testing as a desirable difficulty in the classroom. *Journal of Applied Research in Memory and Cognition*, 3(3):165–170, 2014.

[5] Luiz Augusto de Macêdo Morais e Jorge C. A. Figueiredo e Dalton D. S. Guerrero. Students satisfaction with mastery learning in an introductory programming course. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 25(1):1048, 2014.

[6] Carol S Dweck. *Mindset: The new psychology of success.* Random House, New York, NY, US, 2006.

[7] Joe Feldman. *Grading for equity: what it is, why it matters, and how it can transform schools and classrooms.* Corwin, a SAGE Company, 2019.

[8] James Garner, Paul Denny, and Andrew Luxton-Reilly. Mastery learning in computer science education. *ACM International Conference Proceeding Series*, pages 37–46, 1 2019.

[9] Sarah Hug. How do faculty convey growth mindset in computer science teaching? pages 1285–1285. ACM, 3 2021.

[10] Henry L Roediger III, Adam L Putnam, and Megan A Smith. Ten benefits of testing and their applications to educational practice, 2011.

[11] Antti-Juhani Kaijanaho. Fixed versus growth mindset does not seem to matter much a prospective observational study in two late bachelor level computer science courses. 18, 2018.

[12] Shoshana D Katzman, Jennifer Hurst-Kennedy, Alessandra Barrera, Jennell Talley, Elisabeth Javazon, Mary Diaz, and Mary E Anzovino. The effect of specifications grading on students' learning and attitudes in an undergraduate-level cell biology course. *Journal of Microbiology & Biology Education*, 22(3):e00200–21, 2021.

[13] Maya M Khanna. Ungraded pop quizzes: Test-enhanced learning without all the anxiety. *Teaching of Psychology*, 42(2):174–178, 2015.

[14] Brianna Kuhn. *Get the 'F'Outta Here: Exploring Contract Grading as a Decolonizing and Equity-Minded Assessment Practice in Composition Classrooms.* PhD thesis, San Diego State University, 2021.

[15] Chen-Lin C Kulik, James A Kulik, and Robert L Bangert-Drowns. Effectiveness of mastery learning programs: A meta-analysis. *Review of Educational Research*, 60:265–299, 1990.

[16] Daehan Kwak, Patricia Morreale, Sarah T Hug, Yulia Kumar, Jean Chu, Ching-Yu Huang, J Jenny Li, Paoline Wang, and Juan Li. Evaluation of the use of growth mindset in the cs classroom. 7, 2022.

[17] Noel Lejeune. Contract grading with mastery learning in cs 1, 2010.

[18] Michael Lodi. Does studying cs automatically foster a growth mind-set. 2019.

[19] Joshua Martin, Stephen H. Edwards, and Clfford A. Shaffer. The effects of procrastination interventions on programming project success. pages 3–11. ACM, 8 2015.

[20] Brendan McCane, Claudia Ott, Nick Meek, and Anthony Robins. Mastery learning in introductory programming. pages 1–10. Association for Computing Machinery, 2017.

[21] Laurie Murphy and Lynda Thomas. *Dangers of a Fixed Mindset: Implications of Self-theories Research for Computer Science Education.* 2008.

[22] Julia Nee and Emily Remirez. Engaging with linguistic justice through objectives-based learning and independent research. *Proceedings of the Linguistic Society of America*, 6(2):5108, 2021.

[23] Linda Burzotta. Nilson and Claudia J. Stanny. Specifications grading : restoring rigor, motivating students, and saving faculty time. page 153, 2015.

[24] Caleb O'Malley and Ashish Aggarwal. Evaluating the use and effectiveness of ungraded practice problems in an introductory programming course. In *Proceedings of the Twenty-Second Australasian Computing Education Conference*, ACE'20, page 177–184, New York, NY, USA, 2020. Association for Computing Machinery.

[25] Randal Peters and Tom Buckmiller. Our grades were broken: Overcoming barriers and challenges to implementing standards-based grading. *Journal of Educational Leadership in Action*, 2(2):2, 2014.

[26] Keith Quille and Susan Bergin. Promoting a growth mindset in cs1: Does one size fit all? a pilot study. volume 20, pages 12–18. ACM, 6 2020.

[27] Christian Roberson. Techniques for using specifications grading in computer science. 33:192–193, 6 2018.

[28] Christian Roberson. Techniques for using specifications grading in computer science. *Journal of Computing Sciences in Colleges*, 33(6):192–193, 2018.

[29] Kevin R Sanft, Brian Drawert, and Adam Whitley. Modified specifications grading in computer science: Preliminary assessment and experience across five undergraduate courses. *J. Comput. Sci. Coll.*, 36:34–46, 1 2021.

[30] Linda J. Sax, Jennifer M. Blaney, Kathleen J. Lehman, Sarah L. Rodriguez, Kari L. George, and Christina Zavala. Sense of belonging in computing: The role of introductory courses for women and underrepresented minority students. *Social Sciences*, 7(8), 2018.

[31] Jeffrey Schinske and Kimberly Tanner. Teaching more by grading less (or differently). *CBE—Life Sciences Education*, 13(2):159–166, 2014.

[32] Steven C. Shaffer and Mary Beth Rosson. Increasing student success by modifying course delivery based on student submission data. *ACM Inroads*, 4:81–86, 12 2013.

[33] Jane G Stout and Jennifer M Blaney. "but it doesn't come naturally": how effort expenditure shapes the benefit of growth mindset on women's sense of intellectual belonging in computing. 27:215–228, 2017.

[34] Frode Svartdal, Tove I Dahl, Thor Gamst-Klaussen, Markus Koppenborg, and Katrin B Klingsieck. How study environments foster academic procrastination: Overview and recommendations. *Frontiers in Psychology*, page 3005, 2020.

[35] Omiunota Nelly Ukpokodu. Teachers' reflections on pedagogies that enhance learning in an online course on teaching for equity and social justice. *Journal of Interactive Online Learning*, 7(3):227–255, 2008.

[36] William B Whitten II and Robert A Bjork. Learning from tests: Effects of spacing. *Journal of Verbal Learning and Verbal Behavior*, 16(4):465–478, 1977.

[37] Chris Wilcox and Albert Lionelle. Quantifying the benefits of prior programming experience in an introductory computer science course. volume 2018-Janua, pages 80–85. ACM, 2 2018.

[38] David S. Yeager, Paul Hanselman, Gregory M. Walton, Jared S. Murray, Robert Crosnoe, Chandra Muller, Elizabeth Tipton, Barbara Schneider, Chris S. Hulleman, Cintia P. Hinojosa, David Paunesku, Carissa Romero, Kate Flint, Alice Roberts, Jill Trott, Ronaldo Iachan, Jenny Buontempo, Sophia Man Yang, Carlos M. Carvalho, P. Richard Hahn, Maithreyi Gopalan, Pratik Mhatre, Ronald Ferguson, Angela L. Duckworth, and Carol S. Dweck. A national experiment reveals where a growth mindset improves achievement. *Nature*, 573:364–369, 9 2019.