Evaluating Mastery-oriented Grading in an Intensive CS1 Course

Igor dos Santos Montagner
igorsm1@insper.edu.br
Insper
São Paulo, SP, Brazil

Rafael Corsi Ferrão
rafael.corsi@insper.edu.br
Insper
São Paulo, SP, Brazil

Andrew Kurauchi
andrewTNK@insper.edu.br
Insper
São Paulo, SP, Brazil

Mariana Silva
mfsilva@illinois.edu
University of Illinois
Urbana, IL, USA

Craig Zilles
zilles@illinois.edu
University of Illinois
Urbana, IL, USA

ABSTRACT

Allowing students to re-attempt assessments has been shown to be effective in traditional university-level courses in improving student mastery of course content. In this paper, we analyse an intensive programming introductory experience, where first semester university students' full load is a single semester-long course that teaches the basics of programming and software engineering. We study its use of mastery-based grading: offering five (formative) low-stakes quizzes (with retakes), each focused on a single topic, and five (summative) higher-stakes exams that assess all learning objectives. Our research questions are: (i) "Do second chances help students to increase their performance over time in intensive courses?"; and (ii) "Are second chances effective in reducing stress/-mental load/weight of assessments in intensive courses?". We find that (i) offering second chances on quizzes decreases the number of students at risk of failing before the first exam; (ii) students' proficiency in coding tasks (as measured by exam grades) improve during the semester; and (iii) that our schedule reduces anxiety and mental load for students, but only after students take the first chance.

CCS CONCEPTS

• **Applied computing → Computer-assisted instruction**; **E-learning**.

KEYWORDS

Computing Education, Assessment, Second chance testing, Mastery, Computer-based exams

ACM Reference Format:
Igor dos Santos Montagner, Rafael Corsi Ferrão, Andrew Kurauchi, Mariana Silva, and Craig Zilles. 2024. Evaluating Mastery-oriented Grading in an Intensive CS1 Course . In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630841

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM ISBN 979-8-4007-0423-9/24/03…$15.00
https://doi.org/10.1145/3626252.3630841

1 INTRODUCTION

Mastery/proficiency-oriented grading approaches have attracted a lot of recent attention, in computing and beyond (e.g., [8, 13, 14, 17, 19–21]). These grading approaches typically provide students with multiple attempts to demonstrate proficiency on material, to motivate remediation and in recognition of learning later concepts may be difficult without first mastering early ones. Introductory programming (CS 1) courses seem to be ideal candidates for mastery-oriented grading, because they often have a diversity of incoming student experience [1] coupled with high dependence among the concepts [16].

In this paper, we explore providing multiple attempts in a somewhat unique context, a university's program that has its incoming cohort of students take a single, full-load course during their first semester, which serves as both an introduction to programming and software engineering. Because of the larger number of contact hours, the pace of this course exceeds traditional programming introductions, covering and assessing a typical CS 1 curriculum in the first 7 weeks of the semester. While part of a four-year university, this course has similarities to the experience of coding bootcamps [22] and "block-based" colleges (e.g., [3]) that teach a single subject at a time, in that all of students coursework is focused on a single subject at a time. Conventionally, remediation for the additional assessment attempts is expected largely to occur outside of regular class time. As such, it wasn't obvious that these faster pace courses could see similar benefits from providing additional attempts. Furthermore, the additional testing in an already compressed schedule could be fatiguing and stressful for students.

The introduction of additional attempts was part of a revision motivated by student performance in a previous offering of the course. Analysis of assessment data from that previous semester indicated that students who didn't perform well on the quizzes offered during the Python language instruction ended up dropping the course. In particular, out of the 8 students who had quiz average less than 5 (out of 10), only 2 passed the course.

Three changes were made to the course: (1) students were offered re-take opportunities for these quizzes early the following week, (2) these quizzes and the later summative exams were extended to include short answer questions about Python syntax and semantics in addition to the problem solving programming questions, and (3) additional time was provided for students to review for the summative exams before the software engineering projects were introduced.

This paper attempts to assess the success of these changes. In particular, we explore the following research questions:

RQ1 Do second chances help students to increase their performance over time in intensive courses?

RQ2 Are second chances effective in reducing stress/mental load-/weight of assessments in intensive courses?

We perform a mixed-methods study to investigate these questions. In Sections 2 and 3, we discuss related work and provide a description of the course under study, including the structure of the exams. In Section 4, we discuss our methods, which consist of the collection of assessment scores and qualitative analysis of student interviews. In Section 5, we present results, including that the offered second chances are effective at ensuring the whole cohort passed the course and that the additional exams were not viewed as a burden by students, but rather that they appreciated the opportunity to practice and many students took them, even when they didn't need to for grades.

2 RELATED WORK

Traditionally, in mastery-oriented instruction, students only progress to the next topic after mastering the current one [2, 12], which can be challenging for cohort-oriented courses with a fixed duration. This paper is inspired by a mastery-oriented technique called "second-chance testing". *Second-chance testing* [9] (SCT) where summative assessments are offered exactly twice, which fits in the confines of a conventional cohort-oriented synchronous class.

To our knowledge, second-chance testing has only been studied in settings where students are taking multiple courses in parallel. These studies universally report increases in student grades and generally observe a reduction in failure rates from the addition of second-chance testing [7, 9, 11, 15]. Students like second-chance testing [4, 7]. Students typically study similarly for first-chance exams as if they were "one-shot" exams [4, 10] and study for second-chance exams by remediating material missed on the first-chance exam [4]. If the grading policy doesn't incorporate the first-chance exam score into the final grade, then some students will procrastinate and not adequately prepare for the first-chance exam [5]. The existence of second-chance exams reduces students' self-reported test anxiety, even for students that never elect to take second-chance exams [6].

This paper seeks to explore if second-chance testing is equally effective in intensive courses, where students are spending 80% of their learning time in a single course.

3 BACKGROUND

This research was conducted at a small private 4-year university in Brazil as part of an intensive first-semester course that is mandatory for all Computer Science majors. Students admitted into the Computer Science program devote approximately 24 hours per week during the semester to this single course centered on introduction to programming and software development. This engagement includes six 2-hour in-person lecture sessions and five 1-hour office hours per week. Every cohort, typically consisting of around 40 students, is assigned a dedicated room exclusively for all their activities throughout the semester. Students are actively encouraged to make use of this space for the majority of their study hours. The semester is divided into three parts: (1) Introduction to Programming using Python, (2) Game development using Pygame, and (3) Web development using Django. Students must pass all three parts to pass the course. This study focuses on the first part of the course, which has a duration of 7 weeks.

The learning goals of the Introduction to Programming section are similar to most CS 1 courses, and it is structured in a partially self-paced format, giving students access to all course materials at the start of the semester. However, weekly quizzes are administered at the end of each week, aiming to motivate students to maintain their progress, prevent them from falling behind, and offer a chance to receive feedback regarding their understanding of course content. Therefore, during lecture time, students are expected to work on their own to complete handouts which include a mixture of reading, code tracing, multiple-choice questions, and coding exercises. Instructors mostly act as facilitators during class time, supporting student learning by answering questions and discussing problem solving strategies with small groups of students. In some occasions, instructors may give "traditional" mini-lectures and model problem solving techniques using live coding exercises. At the end of this first seven weeks, students are expected to be able to:

- identify common strategies for solving computational problems
- use conditionals, loops, lists, and dictionaries to represent a computational solution to a problem
- write Python programs to solve simple computational problems

3.1 Course assessment

We assess students' achievements of the learning objectives through three distinct types of assessments: weekly quizzes, comprehensive exams, and a significant end-of-unit programming project. Each assessment type possesses specific characteristics, elaborated upon as follows:

The first five weeks of the course are focused on basic Python constructs and their applications in simple problems. Each week starts with the introduction of a new topic and ends with a corresponding quiz. The quizzes are focused on a single topic (e.g. lists or loops) and only assess the most basic competency levels. Students that are proficient in the learning objectives should expect to get close to 100% in all quizzes. The following topics are introduced in each week:

Q1 - mathematical expressions, variables and functions
Q2 - conditionals (*if*), I/O
Q3 - loops (*while*)
Q4 - lists and iteration (*lists, for*)
Q5 - associative arrays and iteration (*dict*)

Exams include more complex questions and measure a deeper understanding of course content. All five exams included questions testing all content, are proctored, are open notes, and have a similar level of difficulty. Students have the opportunity to take all five exams, but only the three highest scores are used to compute their final score. This averaging of the top three scores is used for three reasons: (1) to account for unintended difficulty variance between the exams, (2) to encourage long term retention of the material through distributed practice, and (3) to increase confidence in the

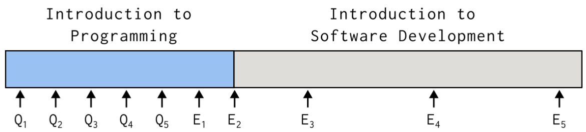

Figure 1: Overview of all CS 1 assessments in the semester. The blue part is exclusively dedicated to Introduction to Programming activities, while the grey part is dedicated to other Software Development activities.

measurement of student ability. Exam questions are graded both manually and automatically using unit tests. The manual grading focuses on code quality and semantic errors and is worth 20% of the total grade. All grading was done by the course instructors.

The weighted score for the CS 1 component is computed as follows:

$$S1 = 0.1QA + 0.55EA + 0.35P$$

where:

QA - quiz average: the average of all quiz grades
EA - exam average: the average of the three best exam scores
P - Individual programming project.

The programming project involves students working individually to produce a fully developed program that covers the entirety of the course material.

A student passes the CS 1 component of the course—the first seven weeks—if $S1$ is larger than or equal to 50% and EA is also larger than or equal to 50%.

The first two exams (*E1* and *E2*) are scheduled at the end of weeks 6 and 7, after all CS 1 topics have been introduced. These last two weeks are dedicated to reviewing and application of Python concepts to increasingly complex problems.

In order to provide additional learning time for students who have not yet achieved the expected proficiency in the CS 1 component of the course, exams *E3, E4*, and *E5* are offered during the weeks dedicated to Introduction to Software Development (see Fig. 1). We expect the new concepts introduced after week 7 to improve students' proficiency in programming as well, so all students can benefit from the extra practice time. To avoid overloading students with practicing their programming skills while learning new Software Development topics, ideally students should have received good scores on the first two exam attempts, reducing the attention required for the other 3 exams. Nonetheless, solutions to all exams are discussed during class and students can redo the questions for reviewing.

3.2 Second-chance schedule

Since quiz grades seemed to be a good predictor of exam performance in a previous offering of the course, the course added second-chance quizzes to encourage students to remediate unlearned quiz material. The schedule for the second-chance quizzes is shown in Figure 2. Each block represents a two-hour class meeting, with blocks of two meetings happening on Monday, Tuesday, and Friday. Beginning on week 2, we offer a retake of the previous quiz on Tuesday morning (3rd block), allowing students to use the weekend and Monday class times for reviewing, if needed. If students decide to skip the retake quiz, they can move on to the new topic covered on that week. The grade Q_i of each quiz was computed as shown below, where $g1_i$ and $g2_i$ are the scores on the first and second chance for quiz i. This policy is known as Weighted Average with Insurance in [9, 10]. If students take only one of the quizzes, that one quiz determines their score.

$$Q_i = \max(0.85\ \max(g1_i, g2_i) + 0.15\ \min(g1_i, g2_i), g1_i)$$

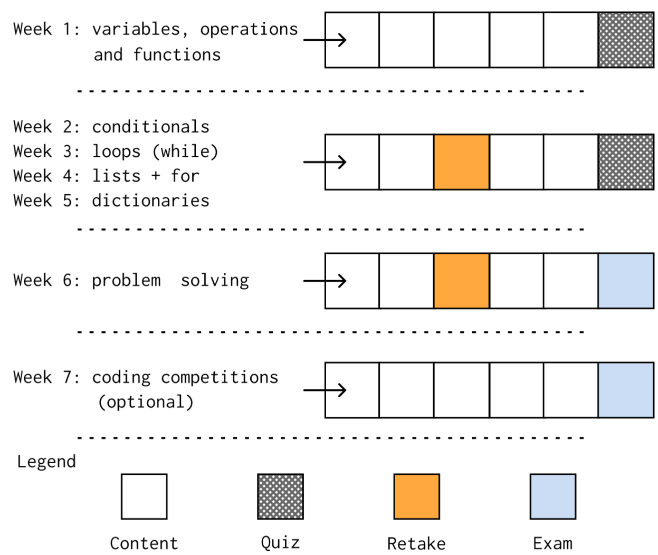

Figure 2: Schedule for the quizzes and first two exams during the CS 1 part of the course. Each square corresponds to a two-hour class meeting block for a total of six meeting times weekly.

3.3 Adding short conceptual questions to quizzes and exams

Each quiz and exam consisted of two sections, each containing one type of question. **Short answer** questions focus only on Python mastery. Question types include code tracing and code fill-in-the-blank code completion questions based on comments and/or input/output pairs. These are created using question generators that add randomization in variable values, inputs, and possible answers. In **Problem Solving** questions, students must interpret a contextualized question statement and write code that passes a set of unit tests.

4 METHODS

This study took place during the Spring 2023 offering of the course; $N = 39$ students were included in the study. The research team was not directly involved in the running of the course but, instead, worked with the instructional team to propose modifications to the course schedule.

We used a mixed methods approach to answer our research questions. Quiz and exam scores were collected, and students were interviewed and the transcripts analyzed qualitatively. The research protocol was reviewed and approved by the institution's Institutional Review Board.

4.1 Quantitative analysis

Anonymized quiz and exam scores were provided to the research team after the semester ended.

For each quiz and exam, we analyzed grades for both **Short Answer** and **Problem Solving** questions. Each quiz was out of 10 points. The two parts of the exam—Short Answer and Problem—were worth 10 points each.

4.2 Qualitative analysis

To gain insights into student perceptions regarding the proposed schedule, we solicited voluntary participation for interviews after the third exam. A total of ten students volunteered and were subsequently interviewed by a member of the research team. The interview protocol included questions about their study methods for quizzes and exams, their understanding of the grading process, their strategy (if any) concerning second chances, and the usefulness of the feedback received after each exam. The interviews were conducted in person and allotted a time span of 30 minutes. The interviews were audio recorded, transcribed, and anonymized.

Grounded Theory [18] was used to analyze the interviews. The interview transcripts were analyzed by two members of the research team. During the open coding stage, the following protocol was followed: the two researchers independently coded two transcripts and then reconciled their code books and the coded transcripts. The remaining transcripts were then independently coded and reconciled two at a time. The final codes were then manually clustered independently to identify themes. The two researchers reconciled this clustering and the themes to mitigate potential interpretation bias.

5 RESULTS

We present our results in three parts: quizzes, exams, and student perceptions. First we sort students into groups according to their decisions regarding taking second-chance quizzes and Exams 4 and 5 and analyze the relationships between student behavior and performance. Second, we use exam grades to evaluate whether there is a significant increase in student grades as they spend more time coding and determine which type of student benefits more from the five exam offerings. Finally, we explore the students' perceptions on the retakes, proposing possible answers to issues raised by the quantitative analysis.

5.1 Second chances on Quizzes

As shown in Figure 3, second-chance quizzes increase students' grades in all quizzes. Because the grading policy is designed with "insurance", students' grades cannot decrease from $g1$ to Q, but it can be seen that the improvement is substantial for most quizzes. The figure slightly exaggerates the benefit of the second chance in that students that skip the first-chance are treated as they received a 0 on the first-chance; this primarily occurs on the Quiz1, due to students joining the course late, and on Quiz5.

On average, about 65% of the students took the second-chance quizzes. This rate was pretty consistent, varying from 49% (Quiz1) to 77% (Quiz2). As previously reported [9, 10], students' decisions to do a retake are influenced by their grades in the first-chance and the

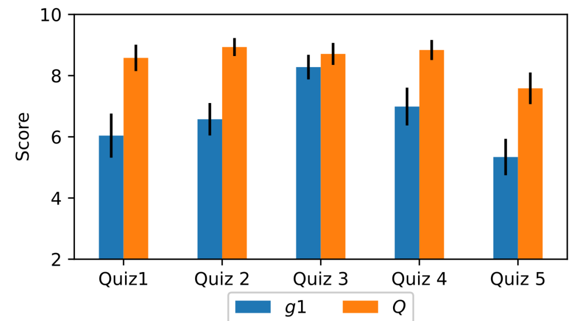

Figure 3: First ($g1$) and final (Q) scores for each quiz

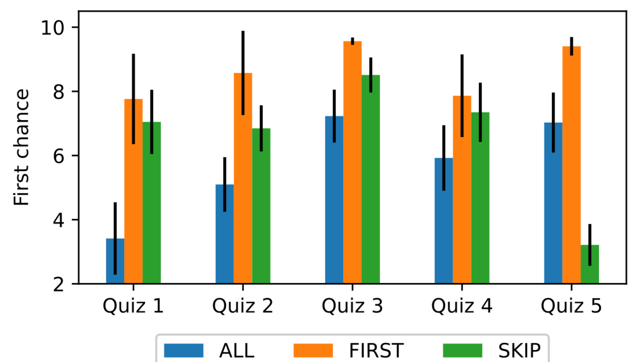

Figure 4: First-chance quiz score by group. ALL performs worst, except in Quiz 5 where most SKIP students didn't take the first chance.

grading policy used, where students with high grades (rationally) are less likely to take the second chance.

To better understand the impact of second chances on different groups of students, we attempted to cluster the students based on their behavior across the five quizzes (both first and second chance). All of the students fell into one of the following three categories:

- **ALL** : they took **all** first chances and retakes ($N = 12$)
- **FIRST** : they took all **first** chances and at least one but not all retakes ($N = 6$)
- **SKIP** : they **skip**ped the first chance at least once ($N = 21$)

In light of **RQ1**, this categorization shows that different students benefit from the second chance exams in different ways. The (**ALL**) appear to be the weakest students, as seen from their first-chance scores in Figures 4. In spite of their poor first-chance performance, they are able to compensate by having significant improvements to their overall grades due to improved scores on the second-chance quizzes, as shown in Figure 5. Their average grade improves to at least 70% in all quizzes, even when their first chance average was below 50%.

In contrast, **FIRST** students are strong performers, generally having high grades on first-chance quizzes. They had less to benefit from second chances and frequently didn't take them (e.g., Quiz 3), but the retakes allowed the **FIRST** students to achieve close to a 100% average on the quizzes.

SKIP are similar to **FIRST** students, but slightly weaker. Their scores are a little harder to interpret because their first-chance

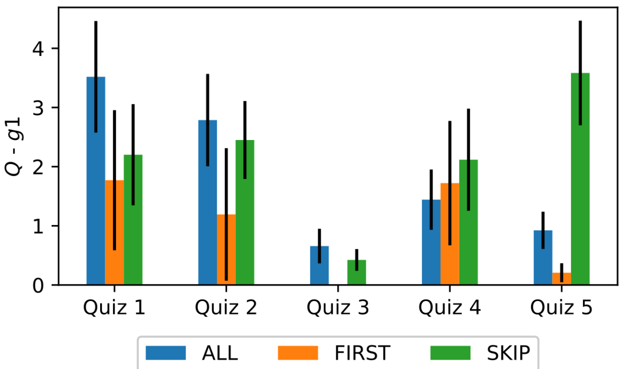

Figure 5: Average benefit from second-chance quiz by group. FIRST and SKIP are able to achieve near perfect scores, except on Quiz 5 where SKIP's performance is lower. ALL is able to substantial improve their scores.

scores include some zeroes for skipping the first chance. This occurs predominantly on Quiz 5, where 16 out of 21 students skipped the first chance. In all quizzes except Quiz 5, **SKIP** students were able to get close to a 100% average; their score on Quiz 5 was much lower, where skipping the first chance was correlated with lower performance. It is unclear, however, which direction the causality goes; did skipping the first chance hurt them or did they skip the first chance because they were struggling with the content. Nevertheless, offering quiz retakes appears to help students improve their scores in all of the topics.

The changes to the course, including the addition of second-chance quizzes reduced the number of "at-risk" students (quiz average below 50%). In the prior offering of the course, eight students were at risk, with all but two of them failing the course. In Spring 2023, only a single student was at risk, and they achieved a passing grade after taking all of the exams.

5.2 Second chances on Exams

Table 1 shows that exam grades for **Short Answer** questions are strong from the beginning (*E1*) and do not change significantly across exams. Scores for the **Problem Solving** questions start lower and trend upward, with decreasing standard deviations. We focus our analysis on the **Problem Solving** questions because that is where the additional exams seem to be useful.

To better understand how our grading system affects students' decisions regarding second chance exams, we clustered the students into four categories based on their average score on **Problem Solving** questions from exams E1 through E3 (we denote this score as s_{13}) and their decision to take or not the following exams:

- **Dev (Developing)**: $s_{13} < 50\%$ - students who took *E4* and/or *E5* to achieve a passing grade ($N = 5$)
- **Acc (Accepting)**: $50\% \leq s_{13} < 80\%$ - students who took *E4* and/or *E5* to increase an already passing grade ($N = 16$)
- **Pro (Proficient)**: $s_{13} \geq 80\%$ - students who took *E4* and/or *E5* to increase an already excellent grade ($N = 14$)
- **Sat (Satisfied)**: students who were satisfied with their current score and elect to not take *E4* and *E5* ($N = 4$)

Table 2 shows the average score for s_{13} for the different categories, as well as the average score s_{14}, which uses the best 3 scores from exams 1 through 4, and the average score s_{15}, which uses the best 3 scores from exams 1 through 5. We observe that the offering of multiple opportunities to take the exams impacts students differently. Students in categories **Pro** and **Sat** did not benefit significantly from exams *E4* and *E5*, since their average score s_{13} was already high. On the other hand, the other two groups (21 out of 39) benefited from the multiple exam offerings, by either achieving a passing score or by substantially improving their already passing scores.

We then examine more carefully the subset of $N = 24$ students who took all exams. We computed a paired t-test between scores on all pairs of exams, with results shown in Table 3. There was a statistically significant increase in scores between exam *E1* and exams *E3*, *E4*, and *E5*. We conclude that one week of additional study time between *E1* and *E2* was not enough to significantly increase these students' proficiency between these two exams. This is further supported by the fact that scores in exams *E3* and *E5* also increased significantly when compared to *E2* and that the increase between *E4* and *E2* is borderline significant. Both exams *E4* and *E5*, which can be viewed as re-takes of exams *E1* and *E2*, benefited students. It should be noted that exam *E4* was an outlier in difficulty, being harder than the other exams. Despite of this increased difficulty, students score statistically significantly better on *E4* than *E1*.

In summary, the two additional exams (*E4* and *E5*) enabled three of the 39 students to go from a failing to passing grade on the exams and improved the grades of many students from borderline to excellent. To retain these benefits while reducing instructor workload, high performing students (*Pro*) may be prevented from taking *E4* and *E5*.

5.3 Student perceptions on second chances

The open coding step of our qualitative analysis of student interviews resulted in 25 codes that were further summarized and grouped into three main themes: **retake decision making, mental state** and **study habits**. For each theme we provide evidence that support our conclusions that second chances indeed reduce stress/mental load/weight of exams in an intensive course (as stated in **RQ2**).

5.3.1 *Retake decision making.*

A large number of students consider that practice is very important for enhancing their knowledge, and the exam is considered a good place for practice, as it is challenging and contextual for them.

Students find it odd not to engage in a graded activity that their peers are participating in, and they are unsure of where to go when the exam is conducted in the lab where they have classes.

> "So it's kind of strange not to take the test. Everyone is doing it and I'm not doing anything else. So I wanted to do it."

They also consider the grading system confusing and don't have much confidence that they can truly opt out of such an activity.

> "I made a mistake on a multiple-choice question, so I wanted to get the maximum grade and decided to

	E1	E2	E3	E4	E5
Short Answer	8.19 (1.11, N=39)	8.49 (1.21, N=36)	8.35 (1.28, N=38)	8.68 (1.11, N=32)	8.70 (1.27, N=27)
Problem Solving	6.82 (2.02, N=39)	7.29 (2.04, N=36)	8.53 (1.92, N=38)	7.73 (1.43, N=32)	8.80 (1.37, N=27)

Table 1: Mean, standard deviation and N for all 5 exams (E1 through E5)

Category	s_{13}	s_{14}	s_{15}
Dev	3.57	4.97	5.70
Acc	6.80	7.43	8.06
Pro	8.86	9.24	9.57
Sat	8.38	8.38	8.38

Table 2: Average scores for the Problem Solving questions in the exams computed by the end of Exams 3, 4 and 5.

	E2	E3	E4	E5
E1	-0.97 (0.34)	**-6.14** (< 0.01)	**-2.69** (0.01)	**-5.41** (< 0.01)
E2		**-4.78** (< 0.01)	-1.93 (0.07)	**-5.12** (< 0.01)
E3			**4.06** (< 0.01)	-1.23 (0.23)
E4				**-5.74** (< 0.01)

Table 3: T-statistic and p-value for paired t-tests between scores for each pair of exams for students that took all exams($N = 24$). Statistically significant results are in bold.

retake it, but I didn't know (...) how to make the calculation to see if it was worth it for me to retake the exam."

5.3.2 Mental state.
Many students believe they need to perform well on their first attempt, and knowing that there are other opportunities does not reduce the initial stress.

> "The first one is what counts. The second one, I'm going to do as well, but we're not going to count on it, and most of the time I did well because I had studied for the first, so the second was a consequence."

However, if poor results are obtained on an assessment, knowing that there is another chance can reduce the students' stress.

> "It was good to have this second chance, because it was not discouraging. I think I even knew some cases of friends who didn't do so well at the beginning, but they're doing well now, and they didn't give up. And for those who are average, it's good because they're still learning and can slip up, and the retake solves it."

Students often wait over two weeks to receive the scores for manually graded portion of the exam. Despite contributing to only 20% of their total grade, students reported being anxious for having to decide whether or not to re-take an exam without knowing their overall score.

> "The grades for the last three exams haven't been released yet. According to the automatic correction, I would get a perfect score, but then I don't know if there will be any deductions for code quality."

5.3.3 Study habits.
The students indicate that the study methods between the initial exam and the retake do not change, and the best approach they identified was to practice and complete all the exercises that are offered, including redoing quizzes and past exams.

> "Before we take a quiz, we had the entire week to work on the handout for a certain topic. [...] And then I just tried to finish the entire handout before taking the quiz on Friday. If I didn't do well on this quiz, it wasn't even because of the content, but more because I got confused and did something wrong, it was simple, the content was not the problem."

The students comment that the quiz helps pace their study and keep up with the material.

> "Here at [redacted] we have much of the 'learn-to-learn' mentality and faculty encourage you to become self-taught. You must always be studying, always up-to-date with the subject matter because, otherwise, it will accumulate, and the faculty won't always be pushing you to study. But with the quizzes, you'll notice that you are always on track, there's always a small test on the schedule, and you need to be studying."

6 CONCLUSIONS

The use of second chances for quizzes and multiple attempts for exams had a positive effect on grades, with many students improving from almost failing to good-to-excellent grades. Quiz retakes in particular were seen as helpful to pace studying and help students to not fall behind. Having multiple attempts at exams allowed students to deal better with stress and anxiety when their grades were less than what they expected. However, this didn't influence their mental state before the first chance. Students also didn't change their study habits after the first exam. Struggling students seemed confident that more practice was necessary and studied more using the same methods, while ones that had better grades didn't put much time into restudying.

Some directions for improvements discussed by students include automated feedback on code quality and a simpler grade system so that students can be more confident about skipping unneeded tests.

7 ACKNOWLEDGEMENTS

We thank the donors of the Insper-UIUC partnership and Insper Computer Science Program for making this research possible (project "Adapting immediate feedback and frequent testing to project-based courses").

REFERENCES

[1] Christine Alvarado, Gustavo Umbelino, and Mia Minnes. 2018. The persistent effect of pre-college computing experience on college CS course grades. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education.*

876–881.
[2] Benjamin Samuel Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13 (1984), 16 – 4.
[3] Colorado College. 2023. The Block Plan: Colorado College. https://www.coloradocollege.edu/basics/blockplan/ [Accessed: 8/16/2023].
[4] Chinedu Emeka, Timothy Bretl, Geoffrey Herman, Matthew West, and Craig Zilles. 2021. Students' Perceptions and Behavior Related to Second-Chance Testing. In *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.
[5] Chinedu Alexander Emeka, David H Smith, Craig Zilles, Matthew West, Geoffrey L Herman, and Timothy Bretl. 2023. Determining the Best Policies for Second-Chance Tests for STEM Students. In *2023 ASEE Annual Conference & Exposition*.
[6] Chinedu Alexander Emeka, Craig Zilles, Matthew West, Geoffrey L Herman, and Timothy Bretl. 2023. Second-Chance Testing as a Means of Reducing Students' Test Anxiety and Improving Outcomes. In *2023 ASEE Annual Conference & Exposition*.
[7] Oscar E Fernandez. 2021. Second chance grading: An equitable, meaningful, and easy-to-implement grading system that synergizes the research on testing for learning, mastery grading, and growth mindsets. *PRIMUS* 31, 8 (2021), 855–868.
[8] Dan Garcia, Armando Fox, Solomon Russell, Edwin Ambrosio, Neal Terrell, Mariana Silva, Matthew West, Craig Zilles, and Fuzail Shakir. 2023. A's for All (As Time and Interest Allow). In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1042–1048. https://doi.org/10.1145/3545945.3569847
[9] Geoffrey Herman, Kavya Varghese, and Craig Zilles. 2019. Second-chance Testing Course Policies and Student Behavior. In *2019 IEEE Frontiers in Education Conference (FIE)*. IEEE, Covington, KY, USA, 1–7. https://doi.org/10.1109/FIE43999.2019.9028490
[10] Geoffrey L. Herman, Zhouxiang Cai, Timothy Bretl, Craig Zilles, and Matthew West. 2020. Comparison of Grade Replacement and Weighted Averages for Second-Chance Exams. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*. ACM, Virtual Event New Zealand, 56–66. https://doi.org/10.1145/3372782.3406260
[11] Sandra M Juhler, Janice F Rech, Steven G From, and Monica M Brogan. 1998. The effect of optional retesting on college students' achievement in an individualized algebra course. *The Journal of experimental education* 66, 2 (1998), 125–137.
[12] Chen-Lin C Kulik and James A Kulik. 1987. Mastery testing and student learning: A meta-analysis. *Journal of Educational Technology Systems* 15, 3 (1987), 325–345.
[13] 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* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 624–630. https://doi.org/10.1145/3545945.3569810
[14] Hui Ma. 2023. Mastery Grading Approach in a Calculus Course. In *Proceedings of the 2023 ASEE Annual Conference and Exposition*.
[15] Jason W Morphew, Mariana Silva, Geoffrey Herman, and Matthew West. 2020. Frequent mastery testing with second-chance exams leads to enhanced student learning in undergraduate engineering. *Applied Cognitive Psychology* 34, 1 (2020), 168–181.
[16] Anthony Robins. 2010. Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education* 20, 1 (2010), 37–71.
[17] Shraddha Sangelkar, Omar M. Ashour, Russel L. Warley, and Oladipo Onipede Jr. 2023. Mastery Learning in Engineering: A Case Study in Statics. In *Proceedings of the 2023 ASEE Annual Conference and Exposition*.
[18] Anselm Strauss and Juliet Corbin. 1994. Grounded theory methodology: An overview. (1994).
[19] Ella Tuson and Tim Hickey. 2022. Mastery Learning and Specs Grading in Discrete Math. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. Association for Computing Machinery, New York, NY, USA, 19–25. https://doi.org/10.1145/3502718.3524766
[20] Ella Tuson and Timothy Hickey. 2023. Mastery Learning with Specs Grading for Programming Courses. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1049–1054. https://doi.org/10.1145/3545945.3569853
[21] Robbie Weber. 2023. Using Alternative Grading in a Non-Major Algorithms Course. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 638–644. https://doi.org/10.1145/3545945.3569765
[22] Graham Wilson. 2017. Building a New Mythology: The Coding Boot-Camp Phenomenon. *ACM Inroads* 8, 4 (oct 2017), 66–71. https://doi.org/10.1145/3132706