Personalized Exam Prep (PEP): Scaling No-Stakes, No-LLM Dialogue-Based Assessments in a Large CS Course

Kelly Cochran
Stanford University
Stanford, CA, USA
kcochran@stanford.edu

Chris Piech
Stanford University
Stanford, CA, USA
piech@cs.stanford.edu

Abstract

Assessing what students know and what work they have done is increasingly challenging in the age of large language models (LLMs) like ChatGPT. At the same time, as class sizes have grown, the CS learning experience has become increasingly depersonalized. Dialogue-based assessments can simultaneously address both of these challenges; but although the benefits of dialogue-based assessments are well-documented, adopting these approaches is often too challenging in lower-division CS courses with hundreds of students. Thus, we sought to develop a dialogue-based intervention that augments existing course assessment structure, accounts for potential student LLM usage, scales to common CS class sizes, and benefits student learning via student-centered design principles. We report our experience implementing "Personalized Exam Prep" (PEP) in two offerings of a 300-student, second-year CS-core course at a North American research-intensive university. PEP consists of one-on-one, TA-guided, problem-solving dialogues, optimized to give students personalized feedback before an upcoming written exam. We describe the logistics of applying this intervention at scale, made possible by custom tooling. We then outline outcomes: positive student feedback, metacognitive impacts, how well PEP-measured performance predicts written exam scores, and the value of PEP as a course-diagnostic for instructors. Finally, we offer insights that may be valuable for large-course instructors updating their assessment structures in a post-LLM world.

CCS Concepts

• **Social and professional topics → Student assessment**.

Keywords

Assessment; LLMs; Large Class Sizes; Oral Exams; Active Learning

ACM Reference Format:
Kelly Cochran and Chris Piech. 2026. Personalized Exam Prep (PEP): Scaling No-Stakes, No-LLM Dialogue-Based Assessments in a Large CS Course. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2026), February 18–21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3770762.3772663

SIGCSE TS 2026, St. Louis, MO, USA

ACM ISBN 979-8-4007-2256-1/2026/02
https://doi.org/10.1145/3770762.3772663

1 INTRODUCTION

In the era of large language models (LLMs), assessment in higher education, and particularly in CS, is evolving [1, 15, 25]. Students can access LLMs capable of correctly solving a majority of lower-division CS course homework problems [9–11, 30, 40], and surveys suggest that student use of LLMs on assignments may be a widespread issue [2, 26, 34, 35, 39]. This problem isn't unique to homework; all unproctored assessments, including online exams, are similarly vulnerable [32, 36]. As excessive student LLM usage on assessments is speculated to impact both learning outcomes and grade distributions, there is a pressing need for instructors to re-evaluate assessment structures and re-align learning goals to better fit our modern AI-integrated reality [6, 8, 15, 25, 26, 35].

Simultaneously, CS course enrollments have been record-high for the past decade [4], creating challenges for instructors in maintaining engagement, providing individualized feedback to students, and applying student-centered pedagogical practices such as active learning [3, 7, 17–19, 27, 37]. LLMs exacerbate these challenges, as consulting LLMs can be more convenient for students than asking course staff for help in large classes [18, 39]. This behavior perpetuates student isolation and worsens learning outcomes, as long as LLMs are suboptimal teachers [5, 7]; every would-be interaction with course staff replaced by an LLM is also a lost opportunity for course staff to maintain an accurate pulse on student progress and course-correct as needed [18] – so the loss of human connection goes both ways. Critically, many recommendations for adapting courses to account for LLMs don't scale easily to large class sizes [20]. Thus, how to make assessment structures more LLM-aware in a way that scales to common CS class sizes is an open question.

Here, we report our experience designing and deploying an intervention to answer this question, which we call "Personalized Exam Prep" or "PEP". To students, PEP is a full-credit-for-participating, no-stakes conversation with a TA that provides students with personalized advice on how to study for an upcoming written exam. To instructors, PEP is a massively parallelized, scalable dialogue-based assessment framework that captures a high-resolution, inherently LLM-free snapshot of learning progress across all students. PEP...

- Bridges the gap between low-stakes, LLM-unregulated homework assignments and high-stakes, no-LLM written exams, without overhauling pre-existing assessment structures
- Combines active learning with prompt, individualized feedback and face-to-face interaction with course staff
- Reflects learning goals revised to focus on students' ability to discuss their work and problem-solving process – skills particularly relevant in a post-LLM world
- Scales feasibly to classes with 300+ students and TA-student ratios as high as 1:24

The contributions of this report are: 1) a reproducible guide to implementing PEP, our design choices, and logistical considerations; 2) a summary of student reactions to PEP and evidence supporting PEP as a meaningful student self-diagnostic; 3) the story of how PEP inadvertently became a course diagnostic for instructors; 4) post-PEP reflections; and 5) shared tooling to enable others to adapt PEP to their courses (`github.com/chrispiech/pep`).

2 RELATED WORK

Extensive prior literature has outlined recommendations for instructors re-structuring courses to counteract potential impacts of unregulated student LLM usage. Although one option is more in-person written exams, there are drawbacks – Prather *et al.* note that increasing exam grade weight can increase student anxiety [25], and even proctored exams are not wholly cheat-proof [16].

Instead, much of the discussion focuses on more "authentic" assessments. Pudasaini *et al.* [26], Sullivan *et al.* [35], and Cotton *et al.* [6] summarize a theme of encouraging assessment re-design to better promote student "reflection" as well as "critical thinking, problem-solving, and communication skills." Several authors re-envision learning goals more aware of how LLM proliferation will affect CS students post-graduation. For example, Lau & Guo [15] and Daun & Brigs [8] suggest that learning goals should reflect the new skillsets software engineers will need in post-ChatGPT workplaces: deeper understanding, problem-solving abilities, and proficiency in discussion and explanation of process. Both Prather *et al.* and Lau & Guo [15] mention oral exams as potential solutions. Gharibyan [12] outlines additional potential benefits of oral exams over written assessments, including the inherently humanizing and personalized nature of face-to-face interaction, the ability for both students and evaluators to ask clarifying questions, and more authentic or deeper measurement of what the student has learned.

Initial adoptions of dialogue-based assessments in CS courses have reported diverse positive outcomes [22]. Ohmann [21] provided foundational evidence that oral exams can compare in difficulty to written exams in CS, with greater instructor-perceived accuracy and positive student reactions. Several studies have demonstrated that oral interviews improve student accountability and, through direct student-teacher interaction, can give students both more detailed feedback and an improved sense of belonging [24, 28, 33]. Sabin *et al.*'s [29] experience conducting virtual oral exams during the pandemic highlighted how real-time oral assessment is inherently LLM-proof. From these sources, two additional themes emerge: first, dialogue-based assessments better align with learning goals focused on real-world skills than written exams. And second, the main impediment to adopting these approaches in CS is large class sizes. SIGCSE 2023's Birds of a Feather session on oral exams [23] illustrates the degree of communal interest in addressing challenges and refining best practices for these assessments.

Our contribution effectively synthesizes the recommendations and reports above into a single intervention, while simultaneously addressing scaling concerns. As a one-on-one dialogue-based assessment where students practice explaining their problem-solving approaches, PEP is both an LLM-free, authentic active learning experience and a realignment of learning goals towards real-world skills. PEP builds on previous work as the first dialogue-based assessment that is both ungraded and positioned specifically in the context of an upcoming written exam, providing PEP with unique benefits. PEP's novelty is further supported by the custom infrastructure we developed, which was critical to scaling to 300+ students.

3 DESIGN

The main PEP experience is a 15-minute one-on-one problem-solving conversation between each student and a TA one week prior to each of the course's two exams. In their PEP session, each student attempts 3-to-4 practice problems calibrated to resemble exam problems with diverse topic coverage. The TA guides the meeting according to a standardized (scripted) flow and provides support if the student gets stuck; simultaneously, they discreetly record a numeric score for how well the student solves each problem. Immediately after the session, the student gets access to a "personalized study plan" that includes a thorough collection of study tips and course topics to review, annotated with recommendations based on the TA's recorded scores.

3.1 The PEP Problem Set

PEP problems were written in the same style and format as the class exams. Cognizant of the short meeting time, we minimized problem word count so that students could focus more on solving the problems than digesting wording. Problems were identical for all students and ordered easy-to-hard and chronologically by lecture coverage, to hopefully prevent students from getting stuck early and not having time to attempt every problem. The final problem, which involved recently covered content, was intended to communicate the expectation that students be up-to-date with lectures.

PEP problem 1 or 2 was always a "homework duplicate", copied from a recent or current assignment. Students could choose to refer to their submitted work when solving this problem in PEP. This problem in particular aimed to assess students' ability to explain their own work and problem-solving process.

For in-person PEP sessions, students received a printed handout of problems at the start (problems were not distributed in advance). Each problem had sufficient scratch space below for students to use if desired, though the TA emphasized that PEP is meant to be conversational, so written or numeric answers are not the goal. For virtual PEP meetings (for remote/ill students), the handout PDF was emailed to students before the meeting start.

3.2 Personalized Study Plans

After their PEP session, students could access a webpage resource with 1) an encouraging message from their PEP TA; 2) a list of course topics with accompanying advice on what to prioritize studying (Figure 1); and 3) a collection of general "Study To-Dos / Pro Tips". The course topic study recommendations were populated programmatically based on TA-recorded scores of student proficiency in the session, where each possible TA score from 1 to 5 maps to a different descriptive recommendation text. The general list of study tips was the same for all students, as unevenly sharing high-level advice seemed unfair. These tips included practical study strategies (e.g. *"do at least two practice exams and time yourself"*) to reduce "hidden curriculum" inequities [13, 31], as well as affective supportive messaging (*"be kind to yourself while studying"*) to provide further at-scale asynchronous support to students with diverse metacognitive and stress-regulating abilities.

4 SCALING UP: LOGISTICS & TOOLING

4.1 Scheduling 300 Meetings

PEP was implemented in a second-year CS course with 330 students enrolled in Fall 2024 and 289 students in Winter 2025. In order to schedule one PEP session for every student, it was essential to have a tool that enabled students to self-schedule. We developed an online sign-up system that was first-come, first-serve and let students freely reschedule their PEP session as much as needed.

To start, we gathered TA availability – with a TA-to-student ratio of approximately 1:24, this meant each TA chose 24 timeslots over a 2-to-3-day window (with day 3 slots being reserved for late student sign-ups; see Section 6.3). For the first PEP trial-run, sessions were 10 minutes each, for 4 hours of total time commitment per TA; switching to 15-minute meetings in later iterations extended this to 6 hours. The tool then uploads all TA timeslots online to populate a drop-down list on a student sign-up webpage. Once a student signs up by selecting a timeslot, the system reserves the timeslot for them. A button allows students to un-reserve the slot and select another as needed. To mitigate potential biases in students' preferences for TAs, TA names are hidden before sign-up and only become visible to signed-up students one day before the session.

Most PEP sessions were in-person to enable face-to-face interaction and prevent student internet use during the session. Because the course had a few remote students, and in anticipation of illnesses, we also made < 10% of available slots virtual over Zoom.

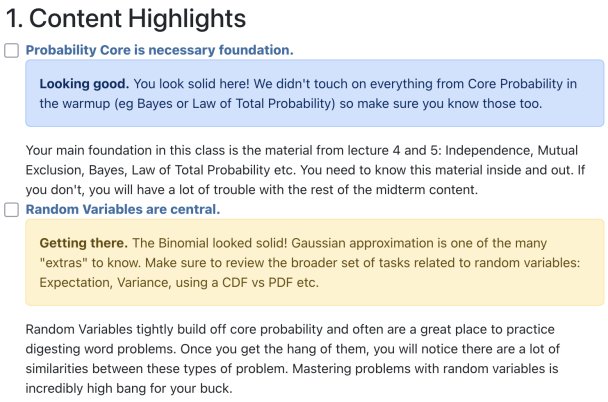

Figure 1: Part of an example personalized study plan, showing recommendations by course topic. The boxed text is set by TA-recorded scores from the student's PEP session.

4.2 TA Script-and-Scoring Interface

The PEP infrastructure also supports TAs during PEP sessions. Each session, the TAs must 1) introduce the student to PEP and walk them through problems in a standardized manner, following a recommended breakdown of minutes spent per problem; 2) pull up the student's submitted work for the "homework duplicate" problem; 3) record how well the student does on each problem; and 4) quickly switch between students for back-to-back sessions. Given the tight time constraint, it was crucial that the TA UI was as streamlined and easy-to-use as possible.

The TA interface consists of a homepage table of all timeslots/students and a page for each student. TAs would begin by navigating from the homepage to the page for their first session/student. This view contains everything a TA needs for a PEP session in an integrated script-and-scoring-system (Figure 2).

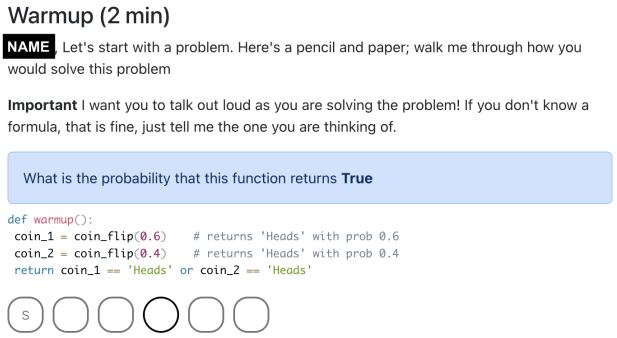

Figure 2: Excerpt from the TA script-and-scoring interface. Here, the 3rd button from right is discreetly selected, representing a score of 3/5; "S" indicates "skipped problem".

The script component is a concise, word-for-word guide of what the TA should tell the student, from key information like *"you are not graded on how well you answer questions"* to tone-setting language like *"This [problem] is meant to be hard. Give it your best, but don't worry if you find it difficult right now."* At the point in the script when the student would attempt each problem, the TA selects from a row of subtle radio buttons to record a score. Score options ranged from 1 to 5, but the TA can also input that the student skipped a problem (usually when they ran out of time). After some practice, TAs knew the script well enough to only rarely glance at their devices and could enter scores quickly and discreetly. Having TAs enter scores during sessions was important, as it meant TAs had zero additional workload from PEP outside of leading sessions.

As TAs engaged conversationally with the student for most of the session, their screen mostly faced away from the student. The exception was for the homework duplicate problem, when the TA would switch their view to "student work only mode" and turn their screen towards the student to let them review their previously submitted work. To display this submitted homework, the tool pulled from our homework app database. Because TAs could toggle their view to only show the student's work, most students never saw any other component of the TA interface. Finally, on each student's page was a button to switch to the correct page for the TA's next session, enabling rapid transitions between students.

4.3 TA Training

Since each student only meets with one TA, TAs can "make or break" PEP experiences. Thus, we aimed to standardize PEP across TAs as much as possible, in line with previous recommendations [21]. The TA interface script streamlined training TAs to stay consistent in time spent per problem, language for managing student expectations, etc. Instructors also communicated PEP "dos" and "don'ts" – for example, *"do provide formulas from class if students ask for them"* and *"don't over-affirm students so much that they decide not to study for the upcoming exam"*. TAs communicated in real-time during PEP using Slack to keep consistent with one another, and as they became more experienced at PEP, best practices evolved. Continuing discussions focused on strategies to handle extreme cases, such as how to go off-script and support students too behind

in the class to attempt any PEP problems, and what a rubric for the numeric scores might look like (future iterations of PEP could benefit from explicit rubrics for TAs).

5 OUTCOMES

5.1 Student Reactions to PEP

Positive quantitative feedback. Students were asked rate their PEP experience while viewing their personalized study plan. Specifically, they responded to the question *"How did we do during your [mid/end-quarter] PEP? This will not be shared with your TA."* using a 5-option scale from Poor to Amazing (Figure 3). For the first PEP trial-run (Fall 2024 mid-quarter), 78% of respondents chose Great or Amazing; for Fall 2024 end-quarter PEP, when we switched to 15-minute sessions, this proportion increased to 88%, and for both iterations in Winter 2025 it stabilized at 92%. Response rates were relatively low (25-31% across all PEPs), and given the question phrasing, these ratings may capture students' impression of their interactions with the TA moreso than approval of PEP overall. Nevertheless, these ratings suggest broadly positive student experiences with PEP.

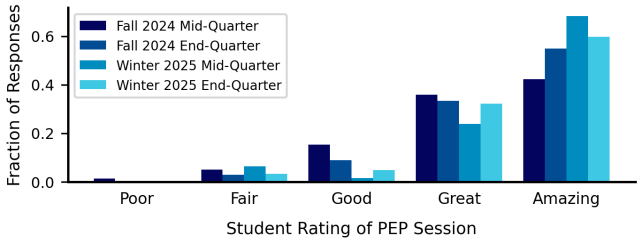

Figure 3: Post-session student responses to the question "How did we do during your PEP?" across four iterations of PEP.

Qualitative feedback: PEP was helpful. We also collected open-ended student feedback on PEP through weekly high-resolution [14] and end-of-quarter anonymous surveys. This feedback most commonly described PEP as *"helpful"*. Students called PEP *"engaging"* and *"interpersonal"* and generally expressed positive sentiment (e.g. *"I really enjoyed the PEP"*). One student specifically characterized PEP as more helpful feedback than discussion sections (*"[in sections,] you don't really get that personalized attention"*); another praised PEP purely from the angle of exam preparation: *"PEP is probably the best strategy to make sure that all students are ready for exams"*. Feedback also included several requests for longer and more PEP sessions in the future. Thus, qualitative feedback also supported PEP as a broadly effective and engaging learning experience, in alignment with previously reported systematic evaluation of the benefits of oral exams in CS courses [22].

Other qualitative feedback: TA preferences, too-short sessions. A recurring neutral comment was students wishing to choose their PEP TA (usually, they wanted their discussion section TA). The most common negative feedback was that sessions were too short: students felt rushed or ran out of time to ask all of their questions. This last point is notable because students also have the option of asking TAs questions at office hours and through an online discussion forum. These students may have seen PEP as a unique opportunity to interact with course staff more interpersonally than through other channels and ask questions they would not have asked otherwise. Finally, one student noted a negative experience where the TA incorrectly rejected their alternative approach to a problem, underscoring the importance of careful TA training for optimizing PEP outcomes.

5.2 Student Performance in PEP

Below, we summarize trends from TA reports on how PEP went, which were collected through informal Slack communications, individual in-person conversations, and team-wide discussions.

TA impressions: high student variability. After the first PEP trial-run, TAs expressed significant surprise at the high variability in proficiency, speed, and expressiveness witnessed across students. Some students trivially solved all problems in half the meeting time, while others struggled to complete the first (easiest) problem within the time. Even among students who solved problems correctly, students varied notably in how well they could articulate explanations of their approach. TAs were trained to simply ask *"but why?"* after students stated an answer without sufficient justification. Surprisingly often, this resulted in the student saying *"I don't know"* or exposing a misconception, highlighting the effectiveness of PEP's dialogue-based format.

Students also varied in their preferred PEP modality. Some embraced the conversational nature of PEP and wrote nothing down, while others preferred working through a problem on a whiteboard to "teach the TA," and others insisted on independently writing out every solution step through to a final numeric answer before talking to the TA. Following prior work's recommendation of maximum flexibility [21], we accepted all student preferences here as long as students orally explained their approach at some point.

The homework duplicate problem. An issue specific to the first trial-run was that PEP revealed a much higher than expected proportion of students who were completing assignments so last-minute that they had not seen the "homework duplicate" problem before PEP. We discuss this outcome further in Section 5.5. Another unanticipated pattern was that across all PEP iterations, a large majority of students strongly preferred to not view their previously submitted work and instead chose to re-solve this problem from scratch. Reasons given varied, but generally, students felt starting over and explaining as they went was more intuitive. Widespread student resistance to explaining their own past work complicated the potential for this PEP problem to catch LLM use on homework. In addition, TAs reported that if a student struggled on the homework duplicate, they usually struggled across the board – and there are many non-LLM explanations for why a student would be struggling overall. Thus, a low PEP score by itself is not concrete evidence of LLM usage. Instead, PEP scores capture student progress towards the learning goal that they should be able to fluently explain their work, regardless of how they produced that work.

5.3 PEP as a Student Self-Diagnostic

The choice to not warn students in advance that PEP sessions were problem-solving assessments gave TAs the opportunity to witness student reactions live, and students commonly reacted by candidly expressing a prediction of how well they would do. Multiple TAs commented that student's apparent confidence levels were poorly calibrated with actual performance. Below, we outline rough trends in how students' apparent confidence related to reception of feedback (self-perceived proficiency at problem-solving) in PEP.

Under-confident students. For students who performed better solving PEP problems than they expected to, TAs described that a common effect of PEP was to *"de-stress."* Some under-confident students would start PEP vocalizing sweeping, generalized concerns about their overall learning progress, but as they worked through problems, this evolved into a more focused self-awareness of specific knowledge gaps, based which PEP problem(s) they found hard. The most drastic cases were students who started PEP expressing complete defeatist un-confidence about the upcoming written exam (e.g. *"I'm cooked [slang meaning doomed]"*), but by making some progress on problems and having a few misconceptions live-corrected by the TA, they found reassurance in knowing more than they thought they did and quickly improving from feedback.

Over-confident students. TAs reported that over-confident students were either apparently unresponsive or over-responsive to feedback from PEP. Unresponsive students justified disregarding PEP feedback by reasoning that the exam was still a full week away, suggesting that PEP's effectiveness as a diagnostic depends on its proximity to high-stakes assessments. On the other hand, overly responsive students appeared hyper-aware of any difficulties encountered while solving problems, verbalizing negative judgmental self-commentary (e.g. *"wow I'm doing badly"* or *"this is embarrassing"*). A major advantage of PEP was that TAs could support students in those moments. When TAs noticed students over-internalizing negative feedback, they could deviate from the script to give a "PEP pep talk." This might look like affective support or re-framing negative feedback as opportunity for course-correction: *"These problems are meant to help you notice knowledge gaps. Getting this feedback now is good, since you have a whole week before the exam to practice."*

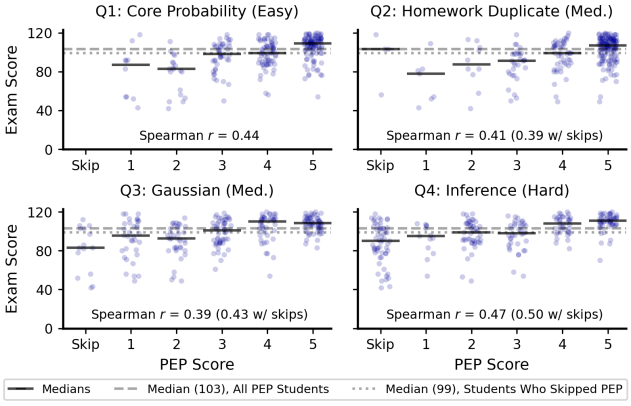

Figure 4: TA-scored student PEP performance per-problem vs. follow-up written exam scores, mid-quarter Winter 2025.

5.4 PEP Scores Predict Written Exam Scores

To support PEP as a diagnostic predictive of written exam performance, we assessed how well TA-recorded student PEP scores correlated with scores on the follow-up written exam (Figure 4). Focusing first on Winter 2025 mid-quarter PEP and the subsequent exam: PEP scores across four problems consistently correlated positively with exam scores ($r = 0.39 - 0.47$, or $0.39 - 0.5$ when skipping a problem is coded as 0). These correlations were stable across problem difficulty and the number of students who skipped a problem, and they were notably higher than the correlation between written exam scores and the average score across problem sets ($r = 0.31$), likely because the median problem set average was high (98.7%).

Fall 2024 mid-quarter correlations varied more ($r = 0.23 - 0.48$, or $r = 0.31 - 0.48$ when skip = 0), with Q3 and Q4 suffering from the shorter PEP session time and extent of student behindness in the course (see Section 5.5). For end-quarter PEP and final exams, PEP-exam correlations were $r = 0.19, 0.49$, and 0.31 for Q1-3, respectively in Fall 2024 and $r = 0.3, 0.44, 0.43$ in Winter 2025. (60% of end-quarter Fall 2024's Q1 scores were 5/5, which led us to swap Q1 for a new problem in Winter.) These trends suggest that when PEP sessions are long enough and PEP problems are not too simple, PEP scores consistently capture meaningful diagnostic signals for both mid-quarter and end-quarter exam outcomes.

Notably, individual datapoints describe several students who performed poorly on PEP but very well on the exam. This could be due to strong responses to PEP feedback, stark preferences for written vs. oral assessments, or an issue with some exam scores.

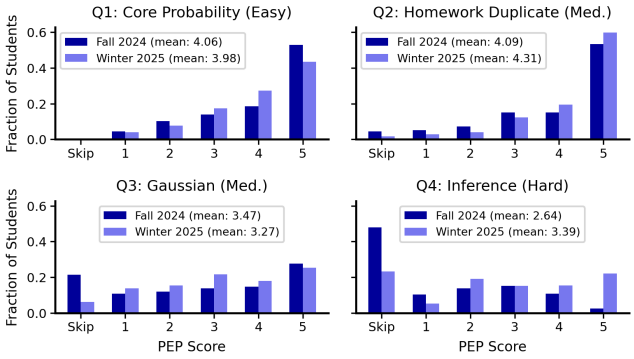

Figure 5: Distribution of student scores for mid-quarter PEP, Fall 2024 (trial-run) vs. Winter 2025 (after course changes).

5.5 PEP as a Course Diagnostic

As mentioned in Section 5.2, PEP captured a snapshot of more highly variable student proficiency than anticipated. In particular, the first trial-run's scores were unexpectedly dominated by how far behind in the class students were. One TA estimated that 40% of students they met in PEP were too behind to attempt past the first PEP problem, and performance on the "homework duplicate" depended heavily on students starting the assignment. TA surprise at these trends was notable because all PEP TAs also led discussion sections and hosted office hours; thus, the signals TAs noticed in PEP were being missed in other student-TA interactions.

This finding led instructors to reflect on course pacing, make changes, and measure the effectiveness of those changes through PEP scores the next quarter (Figure 5). In our data, this comparison is imperfect, as after the first trial-run we made sessions 5 minutes longer and replaced Q4 with a new problem on the same topic which may not be equal difficulty. Nevertheless, from Fall 2024 to Winter 2025, the fraction of students who skipped Q2, Q3, or Q4 (typically students who struggled on Q1-Q2) in mid-quarter PEP more than halved. Scores on Q4 particularly improved, from 2.64 to 3.39; with skips coded as 0, they almost doubled from 1.38 to 2.6.

Thus, PEP was uniquely valuable as a high-resolution measurement of learning goal progress. Where to target course adjustments was guided by TA's rich descriptions of PEP observations, which

were more informative than other observable data for diagnosing the root issue. Unlike homework grades, PEP scores also have the benefit of being free of LLM usage bias; and unlike high-stakes exams, the fact that PEP is ungraded allows reuse of the same problems across course iterations, so scores can be directly compared between quarters to quantify impacts of course adjustments.

6 REFLECTIONS

6.1 To Grade or Not to Grade?

We initially implemented PEP in Fall 2024 as an optional extra-credit opportunity. In Winter 2025, PEP became mandatory, but students earned full credit purely for attending the session. Course staff discussed whether PEP scores should factor into students' grades, but the consensus among TAs was that keeping PEP ungraded was necessary to maintain positive student engagement, candidness, and receptiveness to feedback and reduce student stress.

We speculate that the unique positive outcomes of PEP depend on *both* its ungraded nature and the framing of PEP as exam preparation. PEP's temporal proximity to a written exam, along with messaging that positioned PEP as the ideal exam study support, encouraged student buy-in, especially from more grade-motivated strategic learners. In contrast to other student-TA interactions (i.e. office hours), which are often homework-focused, PEP's association with a summative assessment led to higher-level conversations about overall student progress in the course. Exam study support is also a student-TA interaction space where TAs feel the most "on your side" to students, which may have contributed to positive student experiences. Thus, a major takeaway is the effectiveness of sparse, ungraded, TA-in-the-loop active learning experiences as synergistic precursors to graded higher-stakes assessments.

Messaging to students. To avoid increasing student stress, we intentionally kept messaging about PEP simple. Students were not told in advance that they would be asked to solve problems in PEP, and when students asked, we told them they did not need to prepare for PEP. Despite the no-stakes premise, some students still reported feeling stressed after performing unexpectedly poorly in PEP. Alongside grading considerations, future iterations of PEP can experiment with messaging to better manage student expectations.

Ungraded LLM-free assessment is still valuable. Given the potential for excessive student LLM usage to bias grades, keeping PEP ungraded may seem counter-intuitive. However, without being directly factored into students' grades, PEP scores are still highly valuable datapoints for improving instructor awareness of excessive LLM usage on both homework assignments and written exams. In cases where large discrepancies are observed between performance in PEP vs. on other assessments, instructors have the opportunity to investigate other work from the student for evidence of LLM use (i.e. "unusual" homework submissions, work matching common LLM pitfalls). PEP scores can significantly narrow the search for evidence of excessive LLM usage in large classes – especially those that utilize homework auto-graders – and also provide a high-fidelity orthogonal signal corroborating other evidence of LLM use.

6.2 PEP & Written Exam Calibration

One meta-impact of PEP was its influence on exam calibration. As we developed PEP problems, we were cognizant of how students might calibrate to PEP feedback. We avoided problems so difficult they were discouraging or so easy that students would study less as a result. This required us to carefully anticipate exam difficulty.

Yet inevitably, PEP also influenced exam writing. Once the PEP problem set was created, writing exams unavoidably involved drawing comparisons between candidate exam problems and PEP problems. After all, an exam problem that mimics the structure and difficulty of a PEP problem – which we shared with all students and advertised as "exam-like" – must surely be "fair game." This bi-directional exam-PEP calibration resembled a version of backwards course design [38]: as we calibrated PEP to prepare students for the exam, we also calibrated exams so that students who internalized PEP feedback would be well-positioned for the exam.

6.3 Logistics Lessons Learned

We succeeded at scaling PEP to 300+ students, with the custom tooling being critical. A first-come, first-serve strategy to meeting sign-ups generally worked smoothly, though we also discovered that withholding a few "backup" later timeslots from initial sign-ups helped to create slack in the system for students signing up late with constrained schedules. A larger issue was the rate at which students did not show up to their reserved timeslot (no-shows or last-minute reschedules), which frustrated TAs, who estimated that this happened for 15-25% of sessions. How to better incentivize students to attend PEP on time remains an unresolved question.

Another lesson learned was that students and TAs agreed that 10-minute sessions were too short. As session lengths increased to 15 minutes, PEP stretched from 2 days to 3. Although the TAs showed remarkable capacity to stay personable and alert through 24 sessions in 2 days, they appreciated the flexibility of 3 days. We feared that scheduling some PEP sessions more than 2 days after others would feel unfair to late-session students, but we did not receive this feedback.

7 CONCLUSIONS

In this experience report, we describe the design and outcomes of Personalized Exam Prep (PEP), a scalable framework for one-on-one dialogue assessments of how well students can explain their own problem-solving process. PEP is a sparse add-on to a traditional homework-then-exam assessment structure, targeting the opportunity of an upcoming high-stakes exam to garner student buy-in for a no-stakes active learning experience featuring metacognitive calibration. Custom tooling supported scaling PEP to 300 students with trivial logistic workload for instructors. Though PEP required a large TA team, PEP optimizes TA time for rewarding face-to-face teaching moments, rather than grading or proctoring workloads. PEP can be adapted as-is to any problem-solving course with exams; more generally, our experiential proof-of-concept for developing a scalable, LLM-aware active learning assessment and our post-PEP reflections may be valuable to other instructors looking to modify their course design through similar strategies.

Acknowledgments

We would like to thank the CS109 TA teams whose significant efforts made PEP's success possible; Vanessa Zamora, who supported PEP logistically; and folks at Stanford's Center for Teaching and Learning, including Jamie Imam, Gloriana Trujillo, Nakisha Washington, and Shima Salehi, for their early support of this project.

References

[1] Brett A. Becker, Paul Denny, James Finnie-Ansley, Andrew Luxton-Reilly, James Prather, and Eddie Antonio Santos. 2023. Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Toronto ON Canada, 500–506. doi:10.1145/3545945.3569759

[2] Ritvik Budhiraja, Ishika Joshi, Jagat Sesh Challa, Harshal D. Akolekar, and Dhruv Kumar. 2024. "It's not like Jarvis, but it's pretty close!" - Examining ChatGPT's Usage among Undergraduate Students in Computer Science. In *Proceedings of the 26th Australasian Computing Education Conference*. ACM, Sydney NSW Australia, 124–133. doi:10.1145/3636243.3636257

[3] Kristin Børte, Katrine Nesje, and Sølvi Lillejord. 2023. Barriers to student active learning in higher education. *Teaching in Higher Education* 28, 3 (April 2023), 597–615. doi:10.1080/13562517.2020.1839746 Publisher: SRHE Website _eprint: https://doi.org/10.1080/13562517.2020.1839746.

[4] Tracy Camp, W. Richards Adrion, Betsy Bizot, Susan Davidson, Mary Hall, Susanne Hambrusch, Ellen Walker, and Stuart Zweben. 2017. Generation CS: the growth of computer science. *ACM Inroads* 8, 2 (May 2017), 44–50. doi:10.1145/3084362

[5] Cecilia Ka Yuk Chan and Louisa H. Y. Tsi. 2024. Will generative AI replace teachers in higher education? A study of teacher and student perceptions. *Studies in Educational Evaluation* 83 (Dec. 2024), 101395. doi:10.1016/j.stueduc.2024.101395

[6] Debby R. E. Cotton, Cotton , Peter A., , and J. Reuben Shipway. 2024. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International* 61, 2 (March 2024), 228–239. doi:10.1080/14703297.2023.2190148 Publisher: SRHE Website _eprint: https://doi.org/10.1080/14703297.2023.2190148.

[7] Joe Cuseo. 2007. The Empirical Case Against Large Class Size: Adverse Effects on the Teaching, Learning, and Retention of First-Year Students. *The Journal of Faculty Development* 21, 1 (Jan. 2007), 5–21.

[8] Marian Daun and Jennifer Brings. 2023. How ChatGPT Will Change Software Engineering Education. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. ACM, Turku Finland, 110–116. doi:10.1145/3587102.3588815

[9] Paul Denny, Viraj Kumar, and Nasser Giacaman. 2023. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Toronto ON Canada, 1136–1142. doi:10.1145/3545945.3569823

[10] Michael E. Ellis, K. Mike Casey, and Geoffrey Hill. 2024. ChatGPT and Python programming homework. *Decision Sciences Journal of Innovative Education* 22, 2 (2024), 74–87. doi:10.1111/dsji.12306 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/dsji.12306.

[11] James Finnie-Ansley, Paul Denny, Andrew Luxton-Reilly, Eddie Antonio Santos, James Prather, and Brett A. Becker. 2023. My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises. In *Proceedings of the 25th Australasian Computing Education Conference (ACE '23)*. Association for Computing Machinery, New York, NY, USA, 97–104. doi:10.1145/3576123.3576134

[12] Hasmik Gharibyan. 2005. Assessing students' knowledge: oral exams vs. written tests. *SIGCSE Bull.* 37, 3 (June 2005), 143–147. doi:10.1145/1151954.1067487

[13] Philip Wesley Jackson. 1968. *Life in Classrooms*. Teachers College Press, New York, NY. Google-Books-ID: W46Gu6wwYsQC.

[14] Yunsung Kim and Chris Piech. 2023. High-Resolution Course Feedback: Timely Feedback Mechanism for Instructors. In *Proceedings of the Tenth ACM Conference on Learning @ Scale (L@S '23)*. Association for Computing Machinery, New York, NY, USA, 81–91. doi:10.1145/3573051.3593391

[15] Sam Lau and Philip Guo. 2023. From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1 (ICER '23, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 106–121. doi:10.1145/3568813.3600138

[16] Randall Manteufel, Amir Karimi, and Pranav Bhounsule. 2020. Use of phones and online tutors to cheat on engineering exams. In *2020 Gulf Southwest Section Conference*. ASEE Conferences, Online. doi:10.18260/1-2-370.620-36006

[17] Felix Maringe and Nevensha Sing. 2014. Teaching large classes in an increasingly internationalising higher education environment: pedagogical, quality and equity issues. *Higher Education* 67, 6 (June 2014), 761–782. doi:10.1007/s10734-013-9710-0

[18] P Moodley. 2015. Student overload at university: large class teaching challenges: part 1. *South African Journal of Higher Education* 29, 3 (Jan. 2015), 150 – 167. doi:10.10520/EJC176230

[19] Catherine Mulryan-Kyne. 2010. Teaching large classes at college and university level: challenges and opportunities. *Teaching in Higher Education* 15, 2 (April 2010), 175–185. doi:10.1080/13562511003620001 Publisher: SRHE Website _eprint: https://doi.org/10.1080/13562511003620001.

[20] Keith Nolan, Aidan Mooney, and Susan Bergin. 2015. Facilitating student learning in Computer Science: large class sizes and interventions. College of Computing Technology, Dublin. https://mural.maynoothuniversity.ie/id/eprint/10318/

[21] Peter Ohmann. 2019. An Assessment of Oral Exams in Introductory CS. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, Minneapolis MN USA, 613–619. doi:10.1145/3287324.3287489

[22] Peter Ohmann and Ed Novak. 2025. A Multi-Institutional Assessment of Oral Exams in Software Courses. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 882–888. doi:10.1145/3641554.3701848

[23] Peter Ohmann, Ed Novak, Scott Reckinger, and Shanon Reckinger. 2023. Have You Tried Oral Exams in Your CS Class?. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1250. doi:10.1145/3545947.3573347

[24] Joël Porquet-Lupine and Madison Brigham. 2023. Evaluating Group Work in (too) Large CS Classes with (too) Few Resources: An Experience Report. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Toronto ON Canada, 4–10. doi:10.1145/3545945.3569788

[25] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 108–159. doi:10.1145/3623762.3633499

[26] Shushanta Pudasaini, Luis Miralles-Pechuán, David Lillis, and Marisa Llorens Salvador. 2024. Survey on AI-Generated Plagiarism Detection: The Impact of Large Language Models on Academic Integrity. *Journal of Academic Ethics* (Nov. 2024). doi:10.1007/s10805-024-09576-x

[27] Henry J. Raimondo, Louis Esposito, and Irving Gershenberg. 1990. Introductory Class Size and Student Performance in Intermediate Theory Courses. *The Journal of Economic Education* 21, 4 (1990), 369–381. doi:10.2307/1182537 Publisher: Taylor & Francis, Ltd..

[28] Scott J. Reckinger and Shanon Marie Reckinger. 2021. Oral Proficiency Exams in High-Enrollment Computer Science Courses. In *2021 ASEE Virtual Annual Conference Content Access*. ASEE Conferences, Virtual. doi:10.18260/1-2--37550

[29] Mihaela Sabin, Karen H. Jin, and Adrienne Smith. 2021. Oral Exams in Shift to Remote Learning. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 666–672. doi:10.1145/3408877.3432511

[30] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023. Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses?. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 117–123. doi:10.1145/3587102.3588792

[31] Benson R. Snyder. 1973. *The Hidden Curriculum*. MIT Press. Google-Books-ID: XVScQgAACAAJ.

[32] Alexander Stanoyevitch. 2024. Online assessment in the age of artificial intelligence. *Discover Education* 3, 1 (Aug. 2024), 126. doi:10.1007/s44217-024-00212-9

[33] John A. Stratton. 2021. Enhancing Faculty-Student Interaction in an Undergraduate Algorithms Course Through Group Oral Presentations. In *Proceedings of the 5th Conference on Computing Education Practice*. ACM, Durham United Kingdom, 25–28. doi:10.1145/3437914.3437975

[34] Jérémie Sublime and Ilaria Renna. 2024. Is ChatGPT Massively Used by Students Nowadays? A Survey on the Use of Large Language Models such as ChatGPT in Educational Settings. doi:10.48550/arXiv.2412.17486 arXiv:2412.17486 [cs].

[35] Miriam Sullivan, Andrew Kelly, and Paul Mclaughlan. 2023. ChatGPT in higher education: Considerations for academic integrity and student learning. *Journal of Applied Learning & Teaching* (Jan. 2023). doi:10.37074/jalt.2023.6.1.17

[36] Teo Susnjak and Timothy R. McIntosh. 2024. ChatGPT: The End of Online Exam Integrity? *Education Sciences* 14, 6 (June 2024), 656. doi:10.3390/educsci14060656 Publisher: Multidisciplinary Digital Publishing Institute.

[37] Liz Wang and Lisa Calvano. 2022. Class size, student behaviors and educational outcomes. *Organization Management Journal* 19, 4 (Aug. 2022), 126–142. doi:10.1108/OMJ-01-2021-1139

[38] Grant P. Wiggins and Jay McTighe. 2005. *Understanding by Design*. ASCD. Google-Books-ID: N2EfKlyUN4QC.

[39] Yuankai Xue, Hanlin Chen, Gina R. Bai, Robert Tairas, and Yu Huang. 2024. Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. 331–341. doi:10.1145/3639474.3640076 ISSN: 2832-7578.

[40] Will Yeadon, Alex Peach, and Craig Testrow. 2024. A comparison of human, GPT-3.5, and GPT-4 performance in a university-level coding course. *Scientific Reports* 14, 1 (Oct. 2024), 23285. doi:10.1038/s41598-024-73634-y Publisher: Nature Publishing Group.