2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34799

Oral Proficiency Exams in High-Enrollment Computer Science Courses

Dr. Scott J. Reckinger, University of Illinois at Chicago

Scott J. Reckinger is a Clinical Assistant Professor in the Department of Mechanical and Industrial Engineering and the Department of Computer Science at the University of Illinois at Chicago. He earned a PhD in Mechanical Engineering at the University of Colorado Boulder in 2013. His research interests in computational fluid dynamics are focused on the development and application of advanced numerical methodologies and model development for the study of multi-scale fluid systems, including the growth of hydrodynamic instabilities and the resulting turbulent mixing. Scott teaches courses across many disciplines, including engineering mechanics, introductory programming, probability and statistics, control systems, and professional development. Scott's research interests in engineering education are committed to the advancement of innovative teaching methodologies and pedagogies to improve student learning in inclusive learning environments.

Dr. Shanon Marie Reckinger, University of Illinois at Chicago

Shanon Reckinger is a Clinical Assistant Professor in the department of Computer Science at the University of Illinois at Chicago. She received her PhD in Mechanical Engineering at the University of Colorado Boulder in August of 2011 and an MS degree in Computer Science Education at Stanford University. Her research interests include computational fluid dynamics, numerical methods, and computer science education.

©American Society for Engineering Education, 2021

Oral Proficiency Exams in High Enrollment Computer Sciences Courses

ABSTRACT

Oral proficiency exams were implemented in introductory computer science (CS) courses at a public research minority-serving university. The use of oral exams is motivated by two factors: (1) large CS enrollments; (2) remote learning. The three research questions of this study are (1) to determine if oral exams can provide a way to interact with students one-on-one; (2) to determine if oral exams keep students accountable for obtaining a deep understanding of the course material; and (3) to determine if oral exams are stressful for students. In this study, we implement oral exams in two different courses taught by two different instructors. Students were surveyed at the end of the term to answer the research questions. For research question 1, teaching staff met with students one-on-one three times throughout the semester for enrollments ranging from 100-300 students to complete the oral exams. For research question 2, students reported that oral exams improved their understanding of the material and the oral exams encouraged them to do more independent work throughout the semester. For research question 3, students reported that oral exams were very stressful prior to taking the first one but did not find subsequent oral exams stressful.

MOTIVATION

It has been shown that higher perceived teacher support and school belonging is correlated with less school misconduct [1]. The same study also showed that higher peer attachment is also associated with higher rates of school misconduct. The most common reason students cite for choosing to commit academic misconduct is grade pressure [2]. However, with large and increasing enrollments in introductory programming courses, standardized assessment will be the required for the foreseeable future.

One strategy that has been recommended to combat academic misconduct on out of class assignments is to incorporate in lab exams [3] or regular quizzes [4]. Faculty may perceive incorporation of regular, frequent in-class assessment as establishing more of a passive testing environment, as well as taking up a large proportion of valuable class time. Students may perceive this environment as providing too much assessment with not enough instruction, and/or a stressful, competitive, unwelcoming learning environment, characteristic of many courses within STEM disciplines [5]. As such, these concerns must be addressed directly in the design of any in-class assessments. The intervention presented here uses one-on-one oral coding interview examinations for introductory computer science courses to provide students with extrinsic motivation to learn the material and not resort on short cuts that often fall into the category of academic misconduct.

Previous studies have designed and implemented oral exam assessments in introductory computer science courses. In one study, students rated the difficulty of oral exams equivalent to their traditional written exams [6]. In another study, the oral assessment was designed to mitigate academic misconduct, advance student communication skills, and alleviate negative emotions

associated with traditional exams [7]. Another study concentrated on theoretical computer science courses and found oral exams more effective than written exams as an evaluation method [8].

Additionally, the coding interviews help to improve a sense of belonging in the classroom. There have been many interventions developed in recent years to help improve the sense of belonging for students transitioning to college. One social-belonging intervention is to use carefully written stories from diverse upperclassmen to convey that worries about belonging in a new school are common at first but dissipate with time [9]. Previous experiments using this one-time intervention produced significant results. In one study [10], African American students who received the social-belonging intervention were less likely to respond to daily adversities during the next week with feelings of nonbelonging, engaged in more achievement-promoting behaviors the next week, and earned higher GPAs the next semester than peers. In another study [11], African American students who received the social-belonging intervention earned higher grades throughout the following 3-year period, halving the racial achievement gap. The intervention also increased the percentage of African American students in the top 25% of the class. In another study [12], the social-belonging intervention increased the percentage of students who stayed full-time enrolled in college in their first year from 32% to 43% and increased the percentage of ethnic-minority and first-generation college students who completed the first year full-time enrolled by 4%. In a recent study [13], the long-term effects of the intervention showed greater career satisfaction and success in black adults who had received the treatment 7 to 11 years earlier when compared to the control group. Whereas the majority of these social-belonging interventions have the goal of retention, the goal of the coding interviews is to improve the sense of belonging that will in turn improve students' own learning goals in the classroom.

INTRODUCTION

This paper discusses the design, implementation, analysis, and results of an oral proficiency exam intervention in introductory programming courses. The use of oral exams is motivated by two factors: (1) large CS enrollments and (2) remote learning. Computer science has seen a rapid increase in enrollment over the past 5-10 years, which has resulted in large classes of nearly 300 students. With the continued increase in enrollment, the CS course experience is at a greater risk of impersonality with students lacking a sense of belonging. In some cases, students can go throughout an entire course with little interaction with other students or the instructor. Similarly, with the increase in remote learning due to the COVID-19 pandemic, especially in courses with large enrollments, students are lacking synchronous connections.

An added issue with large enrollments and remote learning is the difficulty of controlling and managing academic misconduct, which may be another symptom of student disconnect from the learning environment. One measure for this effect is the frequency of academic misconduct cases, which has risen in recent semesters. Many instructors have transitioned to asynchronous exams to allow students flexibility when taking a remote learning course. One strategy to mitigate academic misconduct during asynchronous exams is to use randomized questions, which helps to limit the score advantage compared to synchronous exams [14]. However,

asynchronous exams may act to further exacerbate the impersonality of the learning environment.

Thus, the primary aim of the oral proficiency exam intervention is to improve individual student sense of belonging in large-scale introductory programming courses. A secondary aim is to decrease instances of academic misconduct throughout the course by motivating students through increased human interaction.

The three research questions of this study are (1) to determine if oral exams can provide a way to interact with students one-on-one; (2) to determine if oral exams keep students accountable for obtaining a deep understanding of the course material; and (3) to determine if oral exams are stressful for students. In this study, we implement oral exams in two different courses taught by two different instructors. Students were surveyed at the end of the term to answer the research questions.

CODING INTERVIEW INTERVENTION DESIGN

Oral proficiency exams, referred to as *coding interviews* or *in-lab proficiencies*, were implemented in two different introductory CS courses taught by two different instructors at a public research minority-serving university during the Fall 2019 and Spring 2020 semesters. These courses serve as an introduction to programming for non-CS-major engineering undergraduate students and focus on the fundamental concepts of programming and computing, with module topics that include data types, branching, looping, functions, arrays, basic input/output, etc. These concepts are then applied later in the semester to topics such as data analysis (e.g. analyzing the correlation between two data sets), more complicated data structures (e.g. linked lists), basic algorithm development and efficiency (e.g. sorting), sound and image processing, etc. Each course uses a single programming language, either C or MATLAB.

A total of 310 students across two courses and two semesters participated in the coding interview intervention and completed the follow-up survey. The student participants included 222 males and 84 females. Full details on the student demographics (gender, race, and highest parent educational level) for this study, which took place at a minority-serving institution, are given in Table 1. The courses had total enrollments of 122 and 238 in Fall 2019 and 101 and 287 in Spring 2020. The coding interviews were conducted primarily in-person during Fall 2019 and primarily using virtual meeting software during Spring 2020.

Demographics	**n**
All Students	310
Gender	
Male	222
Female	84
Race	
White (non-Hispanic)	103
Asian or Asian American	95
Black or African American	14
Hispanic or Latino	88
Parental Education Level	
Graduate Degree	57
Bachelors Degree	86
Technical or Associates Degree	26
Some College	55
High School	57
Grade School	29

Table 1 – Survey Participant Demographics.

Students completed three coding interview oral examinations throughout the semester during the regularly scheduled laboratory sessions, which are otherwise used for the implementation of coding concepts and development of programming skills through interactive group activities and code-writing exercises. The coding interviews provided an opportunity for each student to meet individually with a Teaching Assistant (TA) or Instructor to discuss the core programming concepts of the course in the context of code that the student wrote for a previous assignment. The TAs were trained to keep the interviews as an informal discussion focused on the coding constructs implemented in the student's code with primary goals as follows:

- To ensure each student is developing fundamental programming skills and to flag those that are falling behind so that additional guidance or support can be directly offered.
- To create a programming-related challenge to every student, regardless of their proficiency level, through oral explanation of coding concepts and one-on-one discussion with a member of the teaching staff.
- To provide students with a real-world experience of having to explain code to a peer or superior, which acts as a component of a mock interview or job task.
- To cultivate confidence in each student and their ability to describe the functionality of their code and their approach to tackling challenging programming tasks.
- To offer an opportunity to the educators (i.e. TAs and instructor) to interact with individual students as a check-in on overall student comprehension and application of the programming concepts.

Keeping the TAs and, thus, the students focused on these goals helped to reduce tension through the coding interview process.

During the lab periods devoted to completing the coding interviews, Graduate TAs, Undergraduate TAs, and the Course Instructor met individually with each student to complete the 10–20-minute oral exam. In order to ensure all students were given the opportunity to complete the coding interview in a single 2-hour lab session, each teaching staff interviewer was assigned at most 8 students to interview. The scheduling was straightforward as each lab section of 20-30 students was already assigned a graduate TA and 2-3 undergraduate TAs to run and support the weekly group programming activities. The instructor for each course attended some, but not all, of the lab sections to partake in the coding interview process in order to be part of the one-on-one assessment.

Throughout the coding interview, the student and teaching staff interviewer worked together at a single computer or through a virtual screen share. A program that the student previously wrote was opened up and was used as the foundation for the interview questions. The interviews required students to analyze, explain, or adapt code based on the interviewer's questions focused on *core proficiencies* of the introductory programming course to that point in the semester. Some sample core proficiencies include the following:

- Understand that code is read and executed **sequentially**.
- Understand how computer programs perform **basic input and output** of information.
- Understand variable declaration of different **data types**
- Understand **branching** using **simple logic**.
- Understand **looping** using **conditional termination**.
- Understand **looping** using **preset incrementation**.

- Understand how to **assign and access array elements**.
- Understand how to **loop through arrays** to perform global calculations.

The following are example coding interview questions that the interviewer may have asked:

- *How does a value get assigned an initial value for the variable* `userNum`*? Is it hardcoded, set by user-input, or a function parameter? Can you show me where in the code it is done and talk me through how you can test your code for different values?*
- *The* `for` *loop is functioning properly as it produces the desired output, but how will the output change if we modify the incrementor statement from* `++i` *to* `i=i+2`*?*
- *Can you please declare an integer array with 10 elements, initialized to values 25-34. Then print/display to screen the value 30 that is stored in the array.*

The evaluation of student performance on the coding interviews was designed to have limited impact on their overall course grade to reduce anxiety for this intervention and to keep the focus on making it an invaluable learning experience. The interviewer assessed the student on each core proficiency using a simple four-point evaluation scale as follows:

0: no proficiency; no concept understanding; difficult to communicate topic
1: very low proficiency; basic concept understanding only; major gap in knowledge and/or application
2: acceptable proficiency; core concept understanding; mostly applies concept successfully
3: full proficiency; mastery of subject; deep concept understanding in a wide variety of applications

The interviewer then evaluated the student on overall proficiency using the same scale. The evaluations were then converted to a *Meets Expectations* vs. *Below Expectations* grading system, where a 2 or 3 converts to *Meets Expectations* and a 0 or 1 converts to *Below Expectations*. The coding interviews did not directly impact the students' course grades, but students were required to complete all coding interviews throughout the semester at a *Meets Expectations* level as a requirement to pass the course. Unlimited retakes were offered to students at the *Below Expectations* level, where students met privately with the instructor or a TA one week after the initial interview. A small minority of students utilized 2 or 3 attempts to successfully complete their oral proficiency exam. On rare occasions, students unable to complete the coding interview at a *Meets Expectations* level after 2 or 3 attempts decided to drop the course. The unlimited retake feature of the intervention is included to mitigate increased stress associated with the oral exams and to help support the belongingness feeling within the learning environment.

After the final oral proficiency examination, students completed a survey that collected demographic information, responses to general questions about belonging, and feedback on whether the coding interviews improved or hindered their learning. The full text of the survey questions is provided in the Appendix.

RESULTS AND DISCUSSION

In order to investigate the impact of the coding interview intervention on student feelings of belongingness and potential for academic misconduct, the following analysis of the survey results is broken into three primary components: (1) student motivation for deeper concept

knowledge and independence in their work; (2) student reported stress levels before and after completing their first interview; and (3) student reported belongingness.

In the analysis to follow, the survey responses are quantified using 5 = Strongly agree, 4 = Agree, 3 = Somewhat agree, 2 = Somewhat disagree, 1 = Disagree, 0 = Strongly disagree, unless otherwise specified.

Motivation for Deeper Content Knowledge and Independent Work

The histograms in Figure 1 show the survey results on student reported motivation to improve understanding of the material and to do their work independently due to the coding interview intervention. The negative skew in the Figure 1(a) histogram and the mean quantified value of 3.07 support a moderate increase in student reported motivation for deeper understanding of course concepts in preparation for the oral proficiency exams. The negative skew in the Figure 1(b) histogram and the mean quantified value of 2.86 support a moderate increase in student reported motivation for independence in their work in preparation for the oral proficiency exams. Furthermore, student responses to these two questions are moderately correlated, with a highly significant (p-value of 1.91×10^{-36}) correlation coefficient of 0.635. Thus, students who report an increase in motivation to achieve a deeper understanding of the material due to the coding interview intervention are also more likely to report approaching their work with higher levels of independence as a result of the oral proficiency exams.

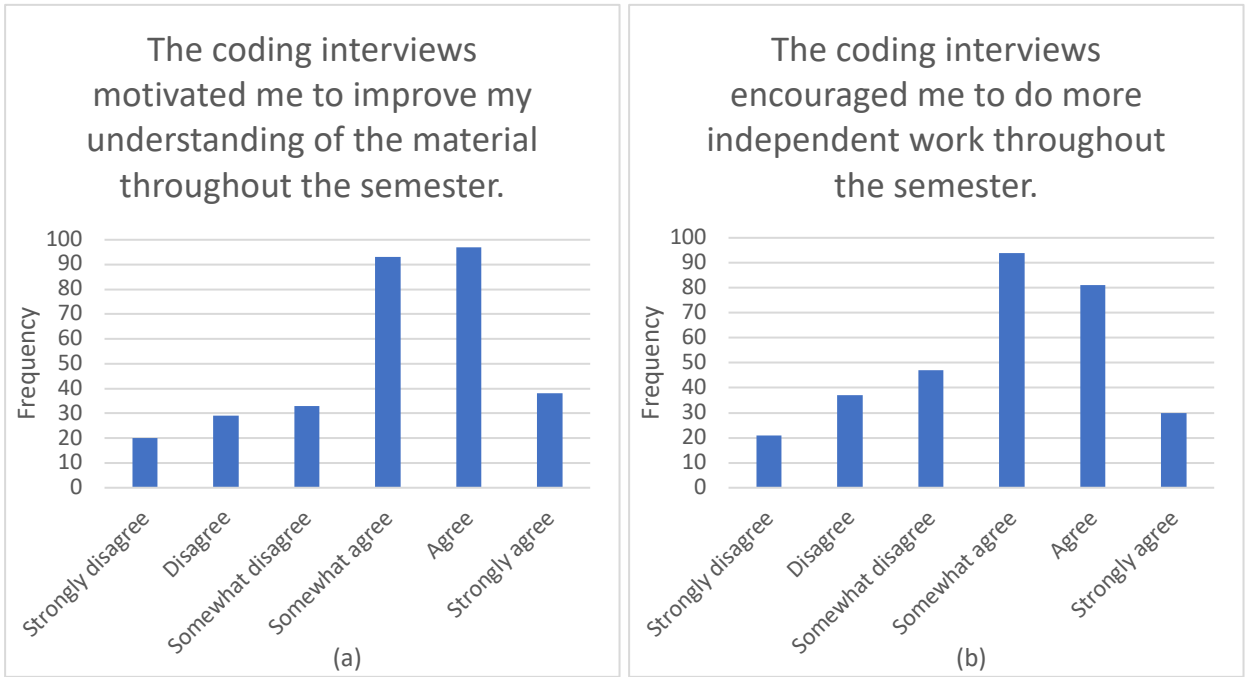


Figure 1 – Aggregated survey results for student reported motivation (a) to obtain deep concept knowledge and (b) to complete work independently.

Stress Level Comparison Before vs. After the Coding Interview

The histograms in Figure 2 show the survey results on student reported stress levels associated with the oral proficiency exams prior to vs. after experiencing the first coding interview. The negative skew in the Figure 2(a) histogram and the mean quantified value of 2.97 support an increased level of reported stress as students prepared for their first coding interview. The positive skew in the Figure 2(b) histogram and the mean quantified value of 1.88 support a substantial decrease in the reported stress level for students after completing the first coding interview and as they prepare for the follow-up interventions. The 1.09 difference in the quantified means for reported stress levels (i.e. reported stess level before – reported stress level after) is statistically significant with negligible p-value for the two-tailed paired sample t-Test.

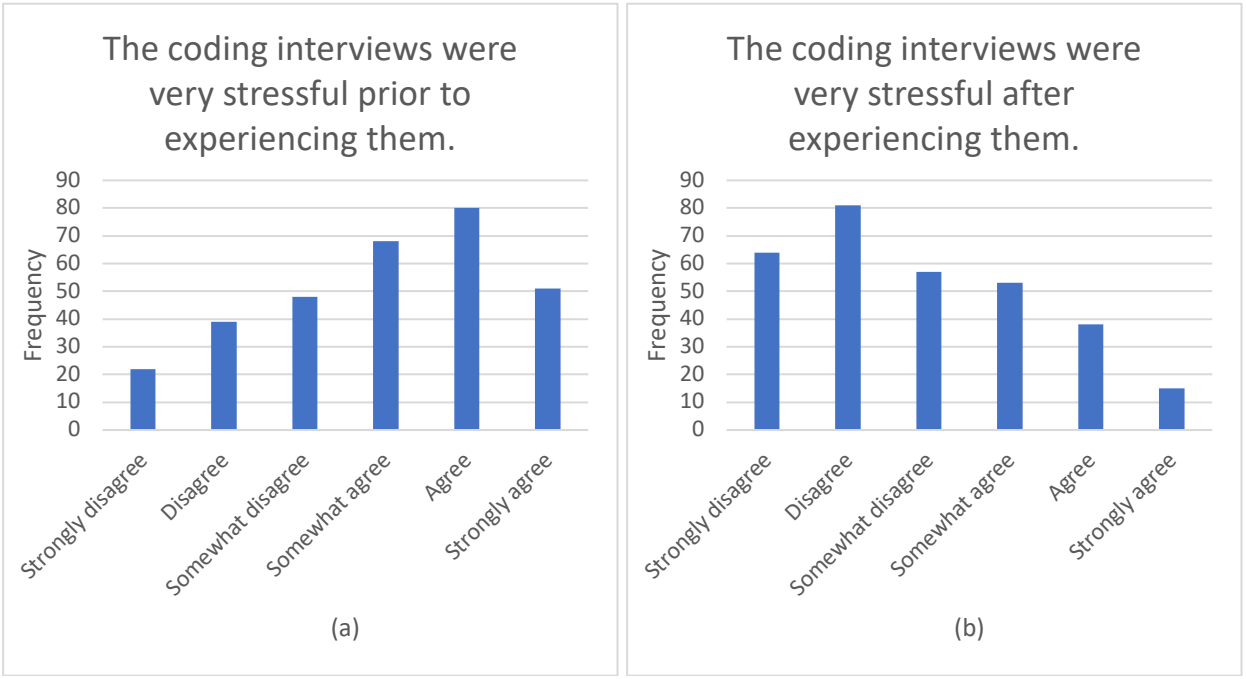


Figure 2 – Aggregated survey results for student reported stress levels (a) before and (b) after completing the first coding interview.

Belongingness

In addition to the survey questions analyzed above that directly address the student feelings toward the oral proficiency exams, data was also collected on student feelings of belongingness in the course and more generally in CS. The survey questions are provided in the Appendix.

The histograms in Figure 3 show the survey results on student reported belongingness variability and overall feelings of belonging in their CS course using image prompts in the survey. Figure 3(a) shows the results for the first image prompt #11, where students select a curvy line to represent their variability in belongingness feelings in the CS course, with A = straight line that has no bends, and E = curvy line with the highest frequency. Figure 3(b) shows the results for the second image prompt #12, where students select among images with a stick figure at different

proximities to the rest of the group to represent their overall feelings of belonging in the CS course, with A = furthest proximity and D = closest proximity. The histogram skews suggest that more students than not experience lower variability in belongingness and higher overall feelings of belonging.

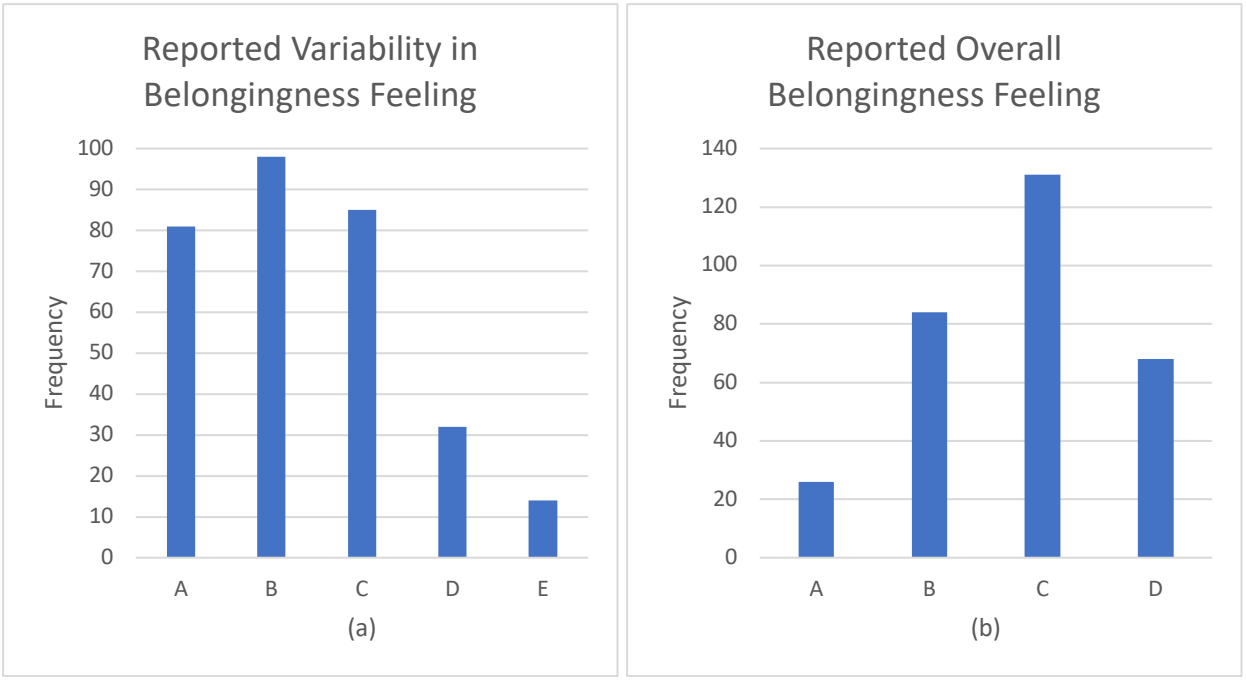


Figure 3 – Aggregated survey results for student reported (a) variability in belongingness associated with survey prompt #11 and (b) overall belongingness feeling associated with survey prompt #12. In (a) the A option is a straight line representing no variability in belongingness, while the E option is a high-frequency curvy line representing extreme variability in belongingness. In (b) the A option shows the stick figure very far from the group representing a low level of overall belongingness, while the D option shows the stick figure in close proximity to the group representing a high level of overall belongingness.

In order to analyze the effects of belongingness feelings on the reported stress levels (both before and after the first oral proficiency exam), motivation for deeper content knowledge, and motivation for work independence among students, two groups were generated for each of the belongingness image prompts. For image prompt #11, the Low Belongingness Variability group includes students that answered A or B (i.e. little to no curviness), while the High Belongingness Variability group includes students that answered C, D, or E (i.e. moderate to high curviness). For image prompt #12, the Low Overall Belongingness group includes students that answered A or B (i.e. far away from the group), while the High Overall Belongingness group includes students that answered C or D (i.e. close proximity to the group). A set of two-sample (assuming equal variances) two-tailed t-Tests were performed on the mean differences between the two groups for each belongingness survey prompt, with results summarized in Table 2. The mean reported stress levels for the High Belongingness Variability group and Low Overall Belongingness group are systematically higher than that for the counterpart groups, both before and after the first coding interview. This result is statistically significant with negligible p-values. However, there is a systematic reduction in reported stress after the first coding interview for all

groups. Thus, among the student participants, the mean reported stress level among students that struggle with belongingness feelings was greater than their peers, but getting through the first coding interview had a large impact on reducing the mean reported stress level for all students going forward in the semester to the later oral proficiency exams.

The disaggregated survey results for reported motivation for deeper content knowledge and independent work summarized in Table 2 across all belongingness groups have means above 2.5 (where 2 = Somewhat disagree and 3 = Somewhat agree). This suggests that the coding oral exam intervention motivates students to obtain deeper understanding of programming concepts and to approach their work with a heightened level of independence.

The only additional statistically significant result related to belongingness level is that the mean reported motivation level for deep content knowledge due to the coding interviews is higher for the High Overall Belongingness group than the Low Overall Belongingness group. Students that struggle to feel like they belong in the CS course report being less motivated by the coding interviews to deeply understand the course material. Thus, improving the inclusivity of CS learning environment may lead to even greater effectiveness of the oral proficiency exam.

	Stress Level Prior	Stress Level After	Deeper Content Knowledge	Independent Work	n
Belongingness Variability	*	*			
Low (A/B)	2.63	1.53	3.15	2.84	179
High (C/D/E)	3.43	2.38	2.96	2.89	131
Difference	-0.80	-0.85	0.19	-0.04	
p-value	2.36E-06	4.16E-07	0.229	0.790	
Overall Belongingness	*	*	*		
Low (A/B)	3.34	2.43	2.69	2.73	111
High (C/D)	2.76	1.58	3.28	2.94	199
Difference	0.57	0.84	-0.59	-0.21	
p-value	1.16E-03	1.18E-06	2.48E-04	0.192	

Table 2 – A comparison of high vs. low reported belongingness variability and overall belongingness, in terms of the mean reported stress levels, both before and after the initial oral proficiency exam, mean reported motivation for deeper content knowledge, and mean reported motivation for work independence. The p-values for two-tailed two-sample (with sample size n) t-Tests on the mean differences are given. The asterisks highlight the statistically significant (i.e. $p < 0.05$) sample mean differences.

CONCLUSIONS

For research question (1), the teaching staff met with students individually for 10-20 minute coding interviews to assess proficiency of the core programming concepts of the course. The oral proficiency exams occurred during regular weekly 2-hour laboratory sessions, three times throughout the semester. The intervention was implemented in two different introductory programming courses, taught by two different instructors, using two different programming

languages, across two semesters, utilizing mixed in-person and virtual meetings, with enrollments ranging from 100-300 students. With the scheduling of the teaching staff (TAs and instructor) support at 8 students/interviewer or less, the coding interviews successfully increased the one-on-one interactions between students and staff.

Managing the workload required to assess student performance and proficiency in introductory programming courses is a difficult task with large and growing enrollments. Whereas recent advancements of effective auto-grader features (such as MathWorks MATLAB Grader) may help with this issue, it may also contribute to the impersonality of the learning environment contributing to a lack of belongingness. Thus, the potential benefits of the coding interview intervention on student belongingness must be weighed against the additional time and effort required.

For research question (2), students reported that the oral proficiency exams motivated them to improve their understanding of the material and encouraged them to do more independent work throughout the semester. The increased motivation for students to take independence in their learning should lead to a reduction in the total volume and level of academic misconduct cases.

In rare situations, the coding interviews confirmed cases flagged by the plagiarism detection software MOSS [15] as cases with high likelihood of code plagiarism and academic misconduct. That is, in cases where students submitted code that they did not write themselves, the coding interview process quickly illuminated gaps in programming concept knowledge and coding syntax that would be required to have written the code as submitted. In these circumstances, the coding interview acts as supplemental evidence for the academic misconduct case. As a primary goal of academic integrity codes is to ensure students achieve course learning objectives, the supplemental evidence provided by the coding interview may lead to quicker academic misconduct resolutions and a stronger refocusing of the student efforts on learning the course material to set themselves up for success in their future.

For research question (3), students reported that oral proficiency exams were very stressful prior to taking the first one but did not find subsequent coding interviews as stressful. Stress levels prior to the first interview were greater for students who reported lower and more variable feelings of belongingness in their computer science course. Despite these obstacles, students with all levels of belongingness reported improvements in content knowledge and independent work due to the oral proficiency exams and even greater stress reduction after the initial coding interview than their majority group peers. A future study will analyze the disaggregated data based on demographics for belongingness deficits and variability in STEM underrepresented groups, stress level differences across demographic groups, and the associated differential effects of the oral proficiency exams on the learning environment.

When implementing oral proficiency exams in introductory programming courses, simply communicating the results of this study with students (i.e. that student stress levels tend to decrease after completing the first coding interview and that students tend to find academic value in the oral proficiency exam process) may help to alleviate anxiety and reduce gaps in belongingness leading to a more productive and inclusive learning environment. The goal should

be to show students the positive impact the coding interview process can have on learning and the invaluable potential impact the practice has on future job opportunities.

ACKNOWLEDGMENTS

We would like to thank the graduate and undergraduate teaching assistants who helped run the coding interviews. We would also like to thank the 748 students in CS 107 and CS 109 who gave their time and effort to their courses and this study. Lastly, we would like to thank the University of Illinois at Chicago and the Department of Computer Science for providing the resources needed to complete this study.

REFERENCES

[1] Demanet, Jannick, and Mieke Van Houtte. "School belonging and school misconduct: The differing role of teacher and peer attachment." *Journal of youth and adolescence* 41.4 (2012): 499-514.

[2] Anderman, Eric M. "The effects of personal, classroom, and school goal structures on academic cheating." *Psychology of academic cheating*. Academic Press, 2007. 87-106.

[3] Barros, João Paulo, Luís Estevens, Rui Dias, Rui Pais, and Elisabete Soeiro. "Using lab exams to ensure programming practice in an introductory programming course." *ACM SIGCSE Bulletin* 35.3 (2003): 16-20.

[4] Fraser, Robert. "Collaboration, collusion and plagiarism in computer science coursework." *Informatics in Education-An International Journal* 13.2 (2014): 179-195.

[5] Gasiewski, Josephine A., M. Kevin Eagan, Gina A. Garcia, Sylvia Hurtado, and Mitchell J. Chang. "From gatekeeping to engagement: A multicontextual, mixed method study of student academic engagement in introductory STEM courses." *Research in higher education* 53.2 (2012): 229-261.

[6] Ohmann, Peter. "An Assessment of Oral Exams in Introductory CS." *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 2019.

[7] 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), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432511

[8] Gharibyan, Hasmik. "Assessing students' knowledge: oral exams vs. written tests." *ACM SIGCSE Bulletin* 37.3 (2005): 143-147.

[9] Walton, Gregory M., and Shannon T. Brady. "The many questions of belonging." (2017).

[10] Walton, Gregory M., and Geoffrey L. Cohen. "A question of belonging: race, social fit, and achievement." *Journal of personality and social psychology* 92.1 (2007): 82.

[11] Walton, Gregory M., and Geoffrey L. Cohen. "A brief social-belonging intervention improves academic and health outcomes of minority students." *Science* 331.6023 (2011): 1447-1451

[12] Yeager, David S., Carissa Romero, Dave Paunesku, Christopher S. Hulleman, Barbara Schneider, Cintia Hinojosa, Hae Yeon Lee et al. "Using design thinking to improve psychological interventions: The case of the growth mindset during the transition to high school." *Journal of educational psychology* 108, no. 3 (2016): 374.

[13] Brady, Shannon T., Geoffrey L. Cohen, Shoshana N. Jarvis, and Gregory M. Walton. "A brief social-belonging intervention in college improves adult outcomes for black Americans." *Science Advances* 6, no. 18 (2020): eaay3689.

[14] Silva, Mariana, Matthew West, and Craig Zilles. "Measuring the Score Advantage on Asynchronous Exams in an Undergraduate CS Course." *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 2020.

[15] Aiken, Alex. "Moss: A system for detecting software plagiarism." *http://www. cs. berkeley. edu/~ aiken/moss. html* (2004).

Appendix: CS Coding Interview Survey Questions

DEMOGRAPHICS:

1. How would you describe your race and/or ethnicity? Mark all the apply:
   - American Indian or Native American
   - Asian or Asian American
   - Hispanic or Latino
   - Black or African American
   - Native Hawaiian or Other Pacific Islander
   - White (non-Hispanic)
   - Other (please specify: _____)

2. What is your gender?
   - Male
   - Female
   - Non-binary
   - Other: _____

3. What is the highest level of education completed by your parent(s) or guardians(s)?
   - Grade school
   - High school
   - Some college
   - Technical or associate's degree
   - Bachelor's degree
   - Graduate degree

BELONGINGNESS:

Read each sentence and then choose the one answer that shows how much you agree with it. There are no right or wrong answers.

4. Overall, how much did you enjoy **this CS course**?
   - Not at all
   - A little bit
   - Some
   - Quite a bit
   - A lot

5. For me, learning to program has been:
   - Extremely Easy
   - Easy
   - Somewhat easy
   - Somewhat difficult
   - Difficult
   - Extremely difficulty

6. When something bad happens, I feel that maybe I don't belong in **this CS course**?
   - Strongly agree
   - Agree
   - Mostly agree
   - Mostly disagree
   - Disagree
   - Strongly disagree

7. Many different kinds of people can be successful computer programmers
    - Strongly agree
    - Agree
    - Mostly agree
    - Mostly disagree
    - Disagree
    - Strongly disagree

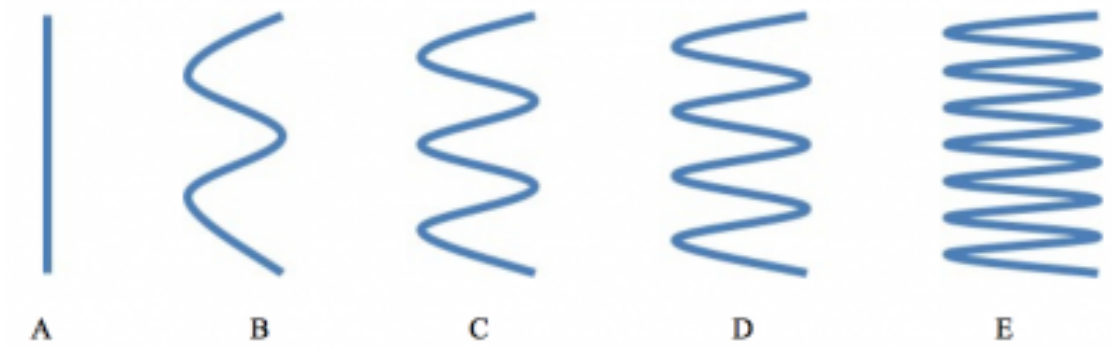
8. It's easy to know if someone is going to be a good computer programmer or not.
    - Strongly agree
    - Agree
    - Mostly agree
    - Mostly disagree
    - Disagree
    - Strongly disagree

9. Do you intend to take computer science (or any computational-focused) courses in the future?
    - Definitely will not
    - Probably will not
    - Might or might not
    - Probably will
    - Definitely will

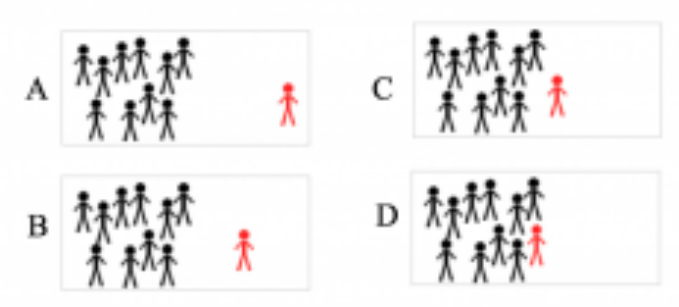
10. How interested are you in majoring or minoring computer science?
    - Extremely interested
    - Interested
    - Somewhat interested
    - Somewhat not interested
    - Not interested
    - Extremely not interested

11. Think about how you feel about yourself at different times. Some people pretty much always feel the same way about themselves. Other people feel differently about themselves at different times.

    The lines below represent a person's feelings of belonging in **this CS course**. The straight line represents someone who always feels the same about how much they belong in **this CS course**. The curved lines represent someone who sometimes feels like they belong in **this CS course** and sometimes feels like they don't belong in **this CS course**. The more curved the line, the more a person's feelings about their belonging in **this CS course** vary. Select the line that best represents how much your feelings about whether you belong in **this CS course** are different at different times.



12. Given the following images, which picture best represents how well you feel you fit in and belong in computer science?



CODING INTERVIEW FEEDBACK

Read each sentence and then choose the one answer that shows how much you agree with it. There are no right or wrong answers.

13. The coding interviews motivated me to improve my understanding of the material throughout the semester:
    - Strongly agree
    - Agree
    - Mostly agree
    - Mostly disagree
    - Disagree
    - Strongly disagree

14. The coding interviews were very stressful prior to experiencing them:
    - Strongly agree
    - Agree
    - Mostly agree
    - Mostly disagree
    - Disagree
    - Strongly disagree

15. The coding interviews were very stressful after experiencing them:
    - Strongly agree
    - Agree
    - Mostly agree
    - Mostly disagree
    - Disagree
    - Strongly disagree

16. The coding interviews encouraged me to do more independent work throughout the semester:
    - Strongly agree
    - Agree
    - Mostly agree
    - Mostly disagree
    - Disagree
    - Strongly disagree

17. Please leave any general comments about the coding interviews here: