Who Attempts Optional Practice Problems in a CS1 Course? Exploring Learner Agency to Foster Mastery Learning

Ashish Aggarwal
ashishjuit@ufl.edu
Department of Engineering Education
University of Florida
Gainesville, Florida, USA

Neelima Puthanveetil
nputhanveetil@ufl.edu
Department of Statistics
University of Florida
Gainesville, Florida, USA

Christina Gardner-McCune
gmccune@ufl.edu
Department of Comp. & Info. Science and Engineering
University of Florida
Gainesville, Florida, USA

ABSTRACT

As enrollments in CS1 courses continue to rise, it has become essential for CS educators to support students with varying learning needs and prior programming experiences. Many experts have pointed to the use of mastery-based learning (MBL), which allows students to develop proficiency by engaging in formative practice problems at their own pace. However, less is known about the characteristics of students who use and benefit from such an approach. CS educators need strong evidence for whether formative practice helps to increase aggregate learning outcomes, especially among students who could gain the most from MBL. In this paper, we are interested in exploring the characteristics of students who engage with formative learning opportunities. We analyze data from 118 students enrolled in a CS1 course who were provided with weekly optional practice quizzes that contained multiple-choice and free-response questions. We used logistic regression to analyze who actually attempted these optional quizzes and found that while gender was not significant, students who do not have prior programming experience (PPE) were more likely to use optional practice than those with PPE. We also conducted a nonparametric two-sample analysis and found that students without PPE engage with optional practice questions to a higher level than students with PPE. Our findings explore the factors that may underpin students' agency and their academic behavior and performance. These results can inform educators on how to scaffold students' learning trajectories by accounting for expected group-based behavioral patterns while utilizing MBL in large CS1 courses.

CCS CONCEPTS

• **Social and professional topics** → *Computational thinking*; *Student assessment*; **CS1**; **Model curricula**.

KEYWORDS

mastery-based learning, CS1, optional practice, formative learning, assessment, pedagogy

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

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569854

ACM Reference Format:
Ashish Aggarwal, Neelima Puthanveetil, and Christina Gardner-McCune. 2023. Who Attempts Optional Practice Problems in a CS1 Course? Exploring Learner Agency to Foster Mastery Learning. In *Proceedings of the 54th ACM Technical Symposium on Computing Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569854

1 INTRODUCTION AND MOTIVATION

As the number of students with mixed abilities and varied prior programming experiences continues to grow in large CS classes, CS educators need to identify effective pedagogical strategies to address the learning needs of these students [7, 8, 33]. Various approaches like pair programming, peer instruction, and process-oriented guided inquiry learning (POGIL) have been found to be effective in improving student engagement and learning outcomes [6, 27, 30, 39]. However, some of them are difficult to implement in larger classes taught in a traditional lecture-based format. Mastery-based learning (MBL) [4], an approach that emphasizes developing proper understanding and fluency of a concept through a formative process, has been gaining more attention in CS education [19, 20, 28].

MBL leverages a pedagogical approach that facilitates self-paced learning through frequent testing with adequate scaffolding and feedback [25]. This method is most commonly implemented with quizzes that help students evaluate whether they are applying course concepts correctly [9]. As students test themselves on various skills, they can also identify their misconceptions and correct them at their own pace, which is not possible in the traditional lecture-based format. While the benefits of using MBL are well known in other disciplines [25], its implementation and effectiveness are just beginning to be studied in the field of CS education [20, 21]. Since learning to program consists of developing fluency with syntax, semantics, and schemas, the time constraint of a lecture may not allow an instructor to discuss a topic comprehensively. In such a context, an MBL approach might be a solution to provide students with opportunities to explore and test themselves on various topics to achieve their desired level of competency [5].

While many researchers have studied the affordances of variational and frequent practice, who engages with it and why remains underexplored. In the case of MBL, especially in the context of large CS classrooms where students can choose to what extent they engage with optional practice questions (OPQ), self-motivated participation may be determined by the estimation of their own learning needs and priorities. In such a setting, it is important to investigate who actually voluntarily engages with OPQ to identify the differences in learners' characteristics based on their behavior and agency. Here 'who' can be classified at many levels based on an individual's learning behavior and intrinsic characteristics like motivation and self-efficacy. However, to the best of our knowledge, no significant study has been conducted on understanding student-group differences based on the use of optional practice problems. This paper attempts to address this gap by analyzing the use of optional formative assessment based on gender and prior programming experience (PPE). We use gender and PPE because they are the factors that have been extensively shown to influence student approaches and performance in CS courses [13, 31, 36]. The results of such an analysis can inform the design of instructional interventions that account for diverse learning behavior patterns.

2 BACKGROUND

Learning by doing or the "doer effect" has long been studied in the learning sciences and learning analytics communities to identify the relationship between the tasks a learner is undertaking and their level of retention and understanding [22]. Koedinger et al. found that in the context of MOOCs, *"students doing more activities learn more than students watching more videos or reading more pages"*, and doing extra activities is effective by a factor of 6 times more than that of extra watching or reading [23]. In another follow-up analysis across four different courses with 12,500 students, they found a positive association between students' engagement with doing extra activities and their learning [24]. To understand the effects of variational practice on students' learning outcomes, Carvalho et al. conducted a study of 1000 students and found that practicing different activities helped improve learning in self-regulated online environments [10]. These studies seem to indicate a relationship between students' engagement with active learning components and their comprehension.

Topics like growth mindset, grading for equity, specifications grading, and MBL have been of growing interest to the SIGCSE community [19, 37]. As educators try to address the issue of high failure rates in programming courses [34] and identify ways to address learner needs, an approach that integrates variational and frequent practice opportunities can help students, as it does not penalize them and provides them with immediate feedback[19]. In CS, instructors can make optional practice problems available either with the use of learning management systems (LMSs) or other tools like CodeWorkout[1] or CodingBat[2]. Estey et al. developed an online tool, BitFit, to support the voluntary practice of programming questions and found that 81% students chose to do at least one question out of over 80 questions available for different topics in a CS1 course [17]. Edwards et al. studied the effect of voluntary practice of short programming exercises on students' performance and, after controlling for ability, found that it improved students' performance in the exams [14]. However, a limitation of this analysis was not accounting for the effect of individual traits that might have affected students' decision to participate in voluntary practice.

Like these studies, many researchers have analyzed either how many students engage with voluntary practice or the impact of voluntary practice on students' performance [16, 32, 38]. But none, to the best of our knowledge, have explicitly studied the characteristics that explain or predict a student's decision to engage with practice problems in CS education. To understand what resources can help students, we must explore group differences in who is more likely to use them and why. Are the students who have PPE more likely to benefit because they have already achieved some level of mastery and are motivated to level up? Or are the students with no PPE more motivated to practice to improve their skills? Also, does its use vary based on gender?

We believe that answers to such questions will help to understand students' motivation better and categorize students based on their characteristics that can be accounted for when addressing students' learning requirements in large programming courses.

3 THEORETICAL FRAMEWORK

Educational and psychological researchers have used various theoretical frameworks to study students' motivation. Some popular frameworks are Bandura's Social Cognitive Theory (SCT) which is based on an individual's self-efficacy, and outcome expectancy [2, 3], and Expectancy Value Theory (EVT) which links a student's expectancy for success at a task to the value of task completion or goal attainment [35].

Deci & Ryan's **Self-Determination Theory (SDT)** explains motivation in the context of an individual's ability to make their own choice [12]. Motivation is a complex construct that can be influenced by intrinsic and extrinsic factors. According to self-determination theory [11], people need:

- ***Autonomy***: Feel in control of their own behaviors and goals. This sense of being able to take direct action that will result in real change plays a major part in helping people feel self-determined.
- ***Competency***: Gain mastery of tasks and learn different skills. When people feel that they have the skills needed for success, they are more likely to take actions that will help them achieve their goals.
- ***Connection or relatedness***: Experience a sense of belonging and attachment to other people.

While many of these theories overlap in their scopes, we guide our analysis based on the Self-Determination Theory. We use SDT as a theoretical framework because we intend to evaluate factors that influence students' decision-making in a context where they are not expected to perform a task but can potentially benefit from it by advancing their learning. The use of optional practice is linked to one's decision and serves as a proxy for their agency based on their sense of autonomy and competence. Given that the optional practice will help students understand and develop mastery of CS concepts, SDT explains the agentic underpinnings of their behavior.

4 RESEARCH QUESTION

With respect to prior work and self-determination theory's applicability in understanding how students make decisions, we are interested in exploring student group-based decisions to attempt optional practice questions. Thus, in the context of a CS1 course for engineering students, we are interested in analyzing: **who uses the optional practice questions (OPQ) based on gender and prior programming experience?**

[1]https://codeworkout.cs.vt.edu/
[2]https://codingbat.com/java

5 COURSE CONTEXT

The data used in this study comes from a two-credit introductory programming course for engineering students that was taught in MATLAB at an R1 university in the US in spring-2021. This course had three sections, all offering an in-person class subsection (capped at 10 students for social distancing purposes) and a larger online class subsection. Each of the three main sections contained around 40 students, and most students registered to be in-person still opted to attend the class virtually.

Since the course style used the flipped classroom model, students were expected to both watch module lectures and complete the pertinent weekly graded quiz before class. These graded quizzes were separate from the optional practice quizzes available to them. Class time was reserved for reviewing the content taught in the video lectures, as well as to complete an in-class activity with the instructor and peers. Students worked on the module homework assignments outside class, which were due at the end of the week.

The key programming concepts covered in the modules included input/output, conditionals, while/for loops, vectors, strings, images, and functions. The course was divided into 14 modules with eight programming-related homework assignments and two exams (a midterm and a final). The last assignment was a cumulative final project.

5.1 Optional Practice Questions (OPQ)

In addition to the regularly required assignments, students were provided with ten optional practice quizzes, which were opened on a weekly basis but were not counted towards their final grades. Once released, students had the choice to practice these quizzes at any time afterward. Every module had 20 questions that were a combination of multiple choice (MCQs) and free response questions (FRQs). These questions were provided through an online web application developed at the institution where this study was conducted. Oftentimes, students in a CS1 course do not master concepts related to syntax and semantics unless they practice and test themselves through various versions of a problem [26, 29]. Thus, this application was designed to provide optional opportunities for repetitive practice on nuanced aspects of a topic that could otherwise generate misconceptions. Most FRQs support output prediction (Figure 1) and code tracing problems, while a large portion of MCQs test general understanding.

In these optional quizzes, students get randomized questions one by one. Before moving from one problem to another, students click the 'Check' button and receive immediate but high-level feedback. If answered correctly, 5% proficiency points are added (since there are 20 questions in each module) to their total proficiency score for that quiz. Otherwise, the system indicates that the question was answered incorrectly and advances to another randomly selected question. After a few new questions are answered, they can reattempt the same question answered incorrectly earlier. In this way, students are not penalized, and they always have the opportunity to practice the questions they initially got incorrect.

The availability of this tool and how to use it was explained both in class and through an email that had instructions on how to use it.

6 DEMOGRAPHICS

There were 121 students across the three sections of the introductory programming course. Due to missing information, three students were excluded from the data set, resulting in a sample size of 118 students. The sections consisted primarily of mechanical and/or aerospace engineering students (61%), civil and/or environmental engineering students (14%), biomedical engineering students (9%), and other students (16%). A total of 60 students (51%) had no prior programming experience, while 58 students (49%) had some form of prior programming experience (PPE). There were 79 (67%) male students and 39 (33%) female students. Academic year-wise, 72 students (61%) were in the first year of their undergraduate program, 29 students (25%) were in their second year, 14 students (12%) were in their third year, 2 students were in the fourth year, and 1 student was in the first year of their graduate program.

Q. What will be the output of the following code?

```
clc; clear;

for ii = 3:3:7
  jj = ii;
  while jj < 5
    jj = jj + 1;
    continue;
    fprintf('%g', jj);
  end
  jj = jj + 2;
  break;
end

fprintf('%g', jj)
```

Correct Answer: 7

Figure 1: Sample free-response question from Module 5

7 METHODOLOGY

Data on students' gender, prior programming experience, and major were collected via an introductory survey during the first week of the course. Data on students' attempts on the optional practice quizzes were retrieved via the web application through which the quizzes were made available to the students. A post-course survey was also given to the students where they were asked about their participation in optional practice quizzes. This study was approved by the university's Institutional Review Board (IRB). Out of the 121 students in the course, three were removed from the data set due to missing information, resulting in a sample size of 118 students.

We used methods applicable to categorical data to conduct our analysis. We coded female students as 0, male students as 1, students without prior programming experience (PPE) as 0, and with PPE as 1.

For our analysis, we also coded the use of optional practice quizzes (Usage) as a binary variable that categorizes every individual student as either a user or not of OPQ. This categorization does not take into account the extent to which they practiced, so any student who engaged and submitted even one question is considered a user. We chose to code Usage as such because we are interested in studying student agency based on the theory of self-determination, which studies the effects of autonomy and the need for competency as sources for deciding whether to perform an action. The

Table 1: Number of students who used OPQ based on gender and PPE

	OPQ Usage = 0	OPQ Usage = 1	Total
Females without PPE	14	11	25
Females with PPE	8	6	14
Males without PPE	18	17	35
Males with PPE	35	9	44
Total	**75**	**43**	**118**

act of creating an account and engaging with even one question is a proxy that differentiates a student from one who never opens the application. Additionally, students' continued engagement and the extent of it may vary based not only on a number of internal and external factors but also on the tool which they are provided with. Thus the extent of engagement can be a function of the design and effectiveness of the tool, which is difficult to discern when studying group-based agentic decisions. Our main focus is on understanding which student group based on gender and PPE takes an action to access and engage with OPQ.

Thus, due to the parametrization of our dependent variables, we used logistic regression models, also known as generalized linear models (GLMs), to understand what factors influence a student to engage with optional practice quizzes.

We use Alan Agresti's An Introduction to Categorical Data Analysis as our primary reference to conduct logistic regression analysis [1]. We also used Excel and R to organize our data and conduct relevant statistical tests.

8 ANALYSIS & RESULTS

Out of a total of 118, 43 students (36.4%) chose to engage with optional practice questions, of which 17 were female and 26 were male; 28 students who used OPQ had no PPE, and 15 did have PPE. A summary of the data we collected has been provided in Table 1.

The general overview of the data of active users from all 10 relevant module quizzes is provided in Table 2. The entries in Min, Max, Median and Avg columns in Table 2 refer to the number of questions attempted. Since students were allowed repeated attempts with no limits, values higher than 20 indicate that there were reattempts of the questions.

The following Figure 2 shows the proportion of students that attempted individual quizzes based on their subgroups determined by gender and PPE.

Our logistic regression models are of the form $log\left(\frac{P(Usage=1)}{P(Usage=0)}\right)$ i.e. the dependent variable is the natural log of the odds that a student uses OPQ given some combination of Gender and PPE. (Notice that it is a common statistical practice to interchange ln and log.) We fit four different models using Gender, PPE, and a combination of these parameters as indicator variables to explain the odds of interacting with OPQ and use both Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) values to select the best model, which are given in Table 3. If a coefficient is statistically significant at an alpha level of .05, we indicate it with * and include its p-value in the third column.

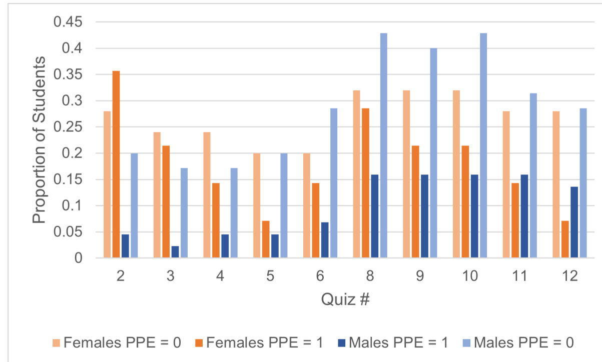

Figure 2: Proportion of students attempting the quizzes relative to their respective group population

Table 2: Number of attempts and the number of students who used OPQ for a particular module quiz

Module#	Topic	#Users	Min(>0)	Max	Median	Avg
2	input/output	21	10	45	28	26.04
3	conditionals	16	7	44	27	24.43
4	while loop	16	5	43	21.5	22.87
5	for loop	15	4	54	19	21.4
6	series-patterns	20	1	77	15.5	19.25
7	**Exam-1**					
8	vectors	34	3	83	19.5	22.82
9	strings	32	3	35	18	18.25
10	images	33	3	38	22	21.09
11	thresholding	27	1	35	23	19.92
12	matrices	24	1	34	12.5	14
13	**Exam-2**					

Does the use of OPQ vary based on Gender or Prior Programming Experience (PPE)?

Based on the results presented in Table 3, we find that the second model where PPE explains the odds that a student uses OPQ has the smallest (i.e., the best) AIC and BIC values and a significant coefficient, so we select $log\left(\frac{P(Usage=1)}{P(Usage=0)}\right) = -.1335 - .9196 \times \mathbf{1}_{\{PPE=1\}}$ as our best model of the four we fit. However, since AIC and BIC serve as criteria to compare models to each other but do not inherently tell us whether the model is actually a good fit for our data, we must perform a series of tests to check the validity of our model. We first conduct a chi-square test of independence to first check whether PPE and Usage are associated. We see that our chi-square test statistic is 5.5113 with 1 degree of freedom, resulting in a p-value of .01889. At an alpha level of .05, **we conclude that PPE and Usage are not independent**.

We also conduct a likelihood ratio test, which compares our chosen model to one including only the intercept term to check if our model with PPE performs better than a model with only an intercept. The likelihood ratio test provides a similar chi-square of 5.5777 on 1 degree of freedom (since the difference between the number of parameters in our model and the intercept-only model

Table 3: Comparison of models based on logistic regression

Model	Maximum Likelihood Estimate of Model	Significant Coefficient(s) & corresponding p-value(s)	AIC Value	BIC Value
1	$.2578 - .4544 \times \mathbf{1}_{\{Gender=1\}}$	None	157.5235	163.0649
2	$-.1335 - .9196 \times \mathbf{1}_{\{PPE=1\}}*$	PPE, p-value = .0202	153.2175	158.758
3	$.04005 - .29866 \times \mathbf{1}_{\{Gender=1\}} - .87050 \times \mathbf{1}_{\{PPE=1\}}*$	PPE, p-value = .0304	154.7041	163.0162
4	$-.24116 + .184 \times \mathbf{1}_{\{Gender=1\}} - .04652 \times \mathbf{1}_{\{PPE=1\}} - 1.125445 \times \mathbf{1}_{\{Gender:PPE\}}$	None	154.4939	165.5767

is 1) with a p-value of .01819. Again at an alpha level of .05, **we conclude that our model with PPE as a parameter performs better than a model with only an intercept**.

We then look at the receiver operating characteristic (ROC) curve and its area under the curve (AUC) to gauge the performance of our model. In the ROC curve shown in Figure 3, sensitivity (true positive rate) is on the x-axis, and the complement of specificity (true negative rate) is on the y-axis, where the y = x or 45-degree line indicated by a dotted blue line represents random guessing of values of the response variable.

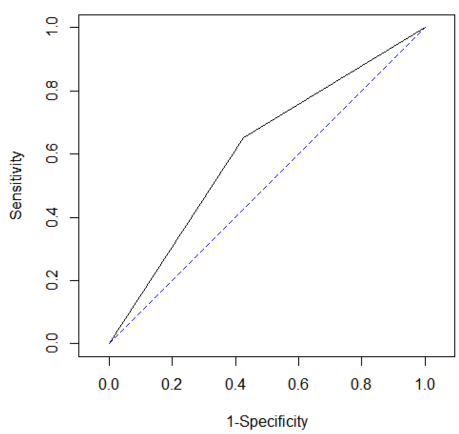

Figure 3: ROC Curve for the model

The AUC is .6122481(> .50), which indicates that including PPE as a parameter in our model does better than random guessing, i.e., if we had no information about a student's prior programming experience.

Lastly, to confirm if PPE is sufficient in explaining the usage of OPQ, we conduct a chi-square goodness of fit test. The null hypothesis is that our current model is sufficient and no more complex models (i.e., additional parameters) are necessary. We get a test statistic of 2.161 on 2 degrees of freedom. The associated p-value is .2562062, which is greater than our alpha level of .05, so we do not reject the null hypothesis. **This implies that including PPE as the sole factor is sufficient to explain the use of OPQ.**

The results from the chi-square tests of association and goodness of fit along with those validating our model's performance, indicate that **PPE plays a predictive role in determining the use of OPQ.**

Recall that our model is:

$$log\left(\frac{P(Usage=1)}{P(Usage=0)}\right) = -.1335 - .9196 \times \mathbf{1}_{\{PPE=1\}}$$

This means that when a student has prior programming experience, the log odds of that student using OPQ decreases by 0.9196 . Equivalently, the odds of a student using OPQ changes by a multiplicative effect, namely $e^{-0.9196}$ = .399 when a student has PPE. Based on our model, ***we can conclude that students without prior programming experience are more likely to use optional practice quizzes than their experienced counterparts, regardless of gender***.

Further analysis on the effect of PPE on students' usage of OPQ when accounting for the number of attempts (level of engagement):

The results from our logistic regression analysis indicate that students without PPE are more likely to use OPQ. Because we coded Usage as a binary variable, we constructed an engagement metric where students' attempts in a module are normalized to be between 0 and 1 (for all the students), and the average of their engagement values across modules is taken to be the new metric. Students may practice the 20 questions in a given module multiple times, perhaps in the same attempt or from using the questions during the pertinent week and around exam time for review. We recognize that normalization (and hence, our engagement scores) are relative since a student who practices the same 20 questions in a module multiple times is given a higher score than a student who only completes the 20 questions once. However, while we are accounting for a more holistic view of their engagement, we also note the merit in distinguishing between these students, as a higher level of engagement or repetitive practicing of a question may serve as a more accurate indicator of a student's learning behavior.

Since the data were highly right–skewed, we use the nonparametric alternative to the two-sample t-test, the Unpaired Wilcoxon Rank Sum test (also known as the Mann-Whitney U test) to check whether the level of engagement is also significant. At an alpha-level of .05, **we have strong evidence (W = 2160, p-value = .004) that students without prior programming experience have higher engagement metric scores than students with PPE**, which is consistent with the interpretation of the coefficients in our logistic regression model.

9 DISCUSSION

There are many factors that can influence both a student's approach towards a CS1 course, and their academic success [15]. While researchers have benchmarked students' performance to evaluate effective instructional strategies and to identify factors that influence performance, less is known about the motivational underpinnings that prompt a student to engage in an activity. The growing complexity related to course logistics in a formal setting and accounting for differences in students' abilities poses a challenge for CS educators. One way to address this issue is to identify strategies that are effective for student subgroups of the course. In order to successfully identify such strategies, we need to analyze the learning behaviors of these subgroup populations. In our study, we explore the use of providing optional practice opportunities to students based on gender and PPE, as they have been widely studied in CS-Ed [13, 31, 36]. We find that PPE has a strong relationship with the use of OPQ. Students without PPE are almost twice as probable to use OPQ than students with PPE. This suggests that CS educators can expect that students with no PPE will likely take advantage of an intervention that they can potentially benefit from. Whether such practice is effective in improving learning outcomes is a contextual question that depends on many factors, especially related to the type of practice being made available, the feedback, the accessibility of the tool, and how the students are examined [14, 17, 32, 38].

To further contextualize our analysis, we wanted to know students' reasons for not engaging with the OPQ. Responses to a survey were collected at the end of the course, which had questions on the use of OPQ. Out of the 106 responses collected, 68 indicated that they did not use OPQ. When asked about the reasons for not attempting the OPQ through a multiple choice, multiple select question, 38 (55%) selected the option that indicated they were doing well in the course and didn't need it, 34 (50%) selected the option that indicated they didn't have the time, 22 (33%) indicated that it was extra overhead to create an account on the website where the OPQ was housed and attempt it there. Two students also indicated that they didn't know that it was available to them.

To understand students' experience of using OPQ, they were asked to describe their experience in the same post-course survey. On analyzing students' responses qualitatively, we found that students discussed various benefits and downsides to the OPQ. Following is a related quote from a student:

> *"Using [OPQ] was very helpful prior to the exam to prepare for them. I did not do the quizzes weekly; I just used them to prepare for the exams."*

Another cluster of students' responses was on the effectiveness of OPQ and how it helped them to improve understanding or to identify misconceptions. The following are some example quotes:

> *"Since [they were] optional and carried no test weight i was less nervous to make mistakes and i could focus on learning and understanding instead of trying to only get the right answer. They are a great study tool."*

> *"..they helped me tremendously in clearing up confusion on syntax. I took these quizzes until I got 100% proficiency because all of the different cases made me build up a better understanding."*

Some students also expressed their dissatisfaction because of the lack of adequate feedback, which might have affected their extent of engagement with OPQ. The following is an example quote:

> *"I did not like the [OPQ] very much due to a single reason: upon getting the answer wrong, I would not get an explanation for the correct answer right away, leaving me wondering and not learning from my mistakes. I feel like practice quizzes are a great idea but I would like to see the answer immediately after the question while it is still fresh in my head."*

Feedback like this gives us insights on potential improvements to improve students' learning experience. While there can be many factors that determine a student's decision to attempt optional practice questions, knowing that students with no prior programming are more probable to use it provides an opportunity for educators to adapt their course design in learning environments in a manner that can help all students.

10 LIMITATIONS AND THREATS TO VALIDITY

For our analysis, the data was collected from a CS1 course for engineering students. Motivation in a CS1 course may differ for students who are non-CS majors as compared to those who are CS majors, and thus may require a different instructional approach [18]. Thus, this may limit the generalizability of our results to other CS1 courses. Another limitation is the use of binary coding for the use of OPQ in our analysis while conducting logistic regression. It is possible that taking into account the extent of engagement may give different results. However, this decision was informed by our research question, which focused on understanding who used the OPQ. In our analysis, we do not take factors like initial performance in the course (Exam-1 score) or other individual traits that can influence their use of OPQ and, thus, limits the validity of our results. Finally, the study assumes that all students were aware of the instructions to access the tool; it is possible that some students may have had technical issues in using it.

11 CONCLUSION & FUTURE WORK

In a CS1 course for engineering students, this paper studies who uses the optional practice questions. Out of a sample of 118 students, we analyze the use of OPQ based on gender and prior programming experience. Results from logistic regression suggest that students who do not have PPE are more likely to use OPQ than students who have PPE. While there have been other studies that have analyzed the effect of voluntary practice on students' performance, these results are the first in examining students' group-based decisions to use OPQ. As CS educators and researchers evaluate different pedagogical interventions to support diverse learning needs, knowing that the learning behavior of student subgroups can help in providing them with tailored resources.

The next step in this stream of analysis is to further consider an individual's prior academic standing (using GPA). It may be possible that students who choose to do OPQ are high-performing students, so taking into account intrinsic ability may provide additional insight into who does OPQ. Given that exam-1 score may impact a student's decision to use OPQ, another potential analysis can be conducted to see if the use of OPQ changes after exam-1. Another follow-up analysis is to study the group-based impact of the use of OPQ on students' performance. The results of the current study suggest that students without PPE engage with OPQ. We would like to explore whether extra practice helps them as compared to other groups in their learning performance.

REFERENCES

[1] Alan Agresti. 2018. *An introduction to categorical data analysis*. John Wiley & Sons.

[2] Albert Bandura. 1991. Social cognitive theory of self-regulation. *Organizational behavior and human decision processes* 50, 2 (1991), 248–287.

[3] Albert Bandura. 2001. Social cognitive theory: An agentic perspective. *Annual review of psychology* 52, 1 (2001), 1–26.

[4] Benjamin S Bloom. 1968. Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment* 1, 2 (1968), n2.

[5] Benjamin S Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher* 13, 6 (1984), 4–16.

[6] Grant Braught, Tim Wahls, and L. Marlin Eby. 2011. The Case for Pair Programming in the Computer Science Classroom. 11, 1, Article 2 (feb 2011), 21 pages. https://doi.org/10.1145/1921607.1921609

[7] Ricardo Caceffo, Guilherme Gama, and Rodolfo Azevedo. 2018. Exploring Active Learning Approaches to Computer Science Classes. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 922–927. https://doi.org/10.1145/3159450.3159585

[8] Dino Capovilla, Marc Berges, Andreas Mühling, and Peter Hubwieser. 2015. Handling Heterogeneity in Programming Courses for Freshmen. In *2015 International Conference on Learning and Teaching in Computing and Engineering*. 197–203. https://doi.org/10.1109/LaTiCE.2015.18

[9] Shana K Carpenter, Steven C Pan, and Andrew C Butler. 2022. The science of effective learning with spacing and retrieval practice. *Nature Reviews Psychology* (2022), 1–16.

[10] Paulo F Carvalho, Elizabeth A McLaughlin, and Kenneth R Koedinger. 2022. Varied practice testing is associated with better learning outcomes in self-regulated online learning. *Journal of Educational Psychology* (2022).

[11] Kendra Cherry. 2021. *How Does Self-Determination Theory Explain Motivation?* https://www.verywellmind.com/what-is-self-determination-theory-2795387

[12] Edward L Deci and Richard M Ryan. 2012. Self-determination theory. (2012).

[13] Rodrigo Duran, Lassi Haaranen, and Arto Hellas. 2020. Gender Differences in Introductory Programming: Comparing MOOCs and Local Courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 692–698. https://doi.org/10.1145/3328778.3366852

[14] Stephen H. Edwards, Krishnan P. Murali, and Ayaan M. Kazerouni. 2019. The Relationship Between Voluntary Practice of Short Programming Exercises and Exam Performance. In *Proceedings of the ACM Conference on Global Computing Education (CompEd '19)*. Association for Computing Machinery, New York, NY, USA, 113–119. https://doi.org/10.1145/3300115.3309525

[15] Stephen H. Edwards, Jason Snyder, Manuel A. Pérez-Quiñones, Anthony Allevato, Dongkwan Kim, and Betsy Tretola. 2009. Comparing Effective and Ineffective Behaviors of Student Programmers. In *Proceedings of the Fifth International Workshop on Computing Education Research Workshop* (Berkeley, CA, USA) *(ICER '09)*. Association for Computing Machinery, New York, NY, USA, 3–14. https://doi.org/10.1145/1584322.1584325

[16] Anthony Estey and Yvonne Coady. 2016. Can Interaction Patterns with Supplemental Study Tools Predict Outcomes in CS1?. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (Arequipa, Peru) *(ITiCSE '16)*. Association for Computing Machinery, New York, NY, USA, 236–241. https://doi.org/10.1145/2899415.2899428

[17] Anthony Estey, Anna Russo Kennedy, and Yvonne Coady. 2016. BitFit: If You Build It, They Will Come!. In *Proceedings of the 21st Western Canadian Conference on Computing Education (WCCCE '16)*. Association for Computing Machinery, New York, NY, USA, Article 3, 6 pages. https://doi.org/10.1145/2910925.2910944

[18] A. Forte and M. Guzdial. 2005. Motivation and nonmajors in computer science: identifying discrete audiences for introductory courses. *IEEE Transactions on Education* 48, 2 (2005), 248–253. https://doi.org/10.1109/TE.2004.842924

[19] Dan Garcia, Connor McMahon, Yuan Garcia, Matthew West, and Craig Zilles. 2022. Achieving "A's for All (as Time and Interest Allow)". In *Proceedings of the Ninth ACM Conference on Learning @ Scale (L@S '22)*. Association for Computing Machinery, New York, USA, 255–258. https://doi.org/10.1145/3491140.3528289

[20] James Garner, Paul Denny, and Andrew Luxton-Reilly. 2019. Mastery Learning in Computer Science Education. In *Proceedings of the Twenty-First Australasian Computing Education Conference (ACE '19)*. Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/3286960.3286965

[21] Mehdi Jazayeri. 2015. Combining Mastery Learning with Project-Based Learning in a First Programming Course: An Experience Report. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. 315–318. https://doi.org/10.1109/ICSE.2015.163

[22] Hassan Khosravi and Kendra M.L. Cooper. 2017. Using Learning Analytics to Investigate Patterns of Performance and Engagement in Large Classes. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 309–314. https://doi.org/10.1145/3017680.3017711

[23] Kenneth R. Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A. McLaughlin, and Norman L. Bier. 2015. Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale* (Vancouver, BC, Canada) *(L@S '15)*. Association for Computing Machinery, New York, NY, USA, 111–120. https://doi.org/10.1145/2724660.2724681

[24] Kenneth R. Koedinger, Elizabeth A. McLaughlin, Julianna Zhuxin Jia, and Norman L. Bier. 2016. Is the Doer Effect a Causal Relationship? How Can We Tell and Why It's Important. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (Edinburgh, United Kingdom) *(LAK '16)*. Association for Computing Machinery, New York, NY, USA, 388–397. https://doi.org/10.1145/2883851.2883957

[25] Chen-Lin C Kulik, James A Kulik, and Robert L Bangert-Drowns. 1990. Effectiveness of mastery learning programs: A meta-analysis. *Review of educational research* 60, 2 (1990), 265–299.

[26] Raymond Lister, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, Beth Simon, and Lynda Thomas. 2004. A Multi-National Study of Reading and Tracing Skills in Novice Programmers. *SIGCSE Bull.* 36, 4 (jun 2004), 119–150. https://doi.org/10.1145/1041624.1041673

[27] Chris Mayfield, Sukanya Kannan Moudgalya, Aman Yadav, Clif Kussmaul, and Helen H. Hu. 2022. POGIL in CS1: Evidence for Student Learning and Belonging. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 439–445. https://doi.org/10.1145/3478431.3499296

[28] Brendan McCane, Claudia Ott, Nick Meek, and Anthony Robins. 2017. Mastery Learning in Introductory Programming. In *Proceedings of the Nineteenth Australasian Computing Education Conference* (Geelong, VIC, Australia) *(ACE '17)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3013499.3013501

[29] Michael McCracken, Vicki Almstrum, Danny Diaz, Mark Guzdial, Dianne Hagan, Yifat Ben-David Kolikant, Cary Laxer, Lynda Thomas, Ian Utting, and Tadeusz Wilusz. 2001. A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-Year CS Students. In *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education* (Canterbury, UK) *(ITiCSE-WGR '01)*. Association for Computing Machinery, New York, NY, USA, 125–180. https://doi.org/10.1145/572133.572137

[30] Charles Mcdowell, Linda Werner, Heather Bullock, and Julian Fernald. 2006. Pair programming improves student retention, confidence, and program quality. *Commun. ACM* 49 (08 2006), 90–95. https://doi.org/10.1145/1145293

[31] Keith Nolan, Aidan Mooney, and Susan Bergin. 2019. An Investigation of Gender Differences in Computer Science Using Physiological, Psychological and Behavioural Metrics. In *Proceedings of the Twenty-First Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '19)*. Association for Computing Machinery, New York, NY, USA, 47–55. https://doi.org/10.1145/3286960.3286966

[32] Caleb O'Malley and Ashish Aggarwal. 2020. Evaluating the Use and Effectiveness of Ungraded Practice Problems in an Introductory Programming Course. In *Proceedings of the Twenty-Second Australasian Computing Education Conference (ACE'20)*. Association for Computing Machinery, New York, NY, USA, 177–184. https://doi.org/10.1145/3373165.3373185

[33] Keith Quille and Susan Bergin. 2019. CS1: how will they do? How can we help? A decade of research and practice. *Computer Science Education* 29, 2-3 (2019), 254–282.

[34] Christopher Watson and Frederick W.B. Li. 2014. Failure Rates in Introductory Programming Revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (Uppsala, Sweden) *(ITiCSE '14)*. Association for Computing Machinery, New York, NY, USA, 39–44. https://doi.org/10.1145/2591708.2591749

[35] Allan Wigfield. 1994. Expectancy-value theory of achievement motivation: A developmental perspective. *Educational psychology review* 6, 1 (1994), 49–78.

[36] Chris Wilcox and Albert Lionelle. 2018. Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 80–85. https://doi.org/10.1145/3159450.3159480

[37] Ursula Wolz, Gail Carmichael, Dan Garcia, Bonnie MacKellar, and Nanette Veilleux. 2020. To Grade or Not To Grade. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 479–480.

[38] Jiayi Zhang, Taylor Cunningham, Rashmi Iyer, Ryan Baker, and Eric Fouh. 2022. Exploring the Impact of Voluntary Practice and Procrastination in an Introductory Programming Course *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 6 pages. https://doi.org/10.1145/3478431.3499350

[39] Daniel Zingaro. 2014. Peer Instruction Contributes to Self-Efficacy in CS1. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) *(SIGCSE '14)*. Association for Computing Machinery, New York, NY, USA, 373–378. https://doi.org/10.1145/2538862.2538878