2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #33360

Exploring the Effect of Quiz and Homework Submission Times on Students' Performance in an Introductory Programming Course in a Flipped Classroom Environment

Leslie Harvey III, University of Florida

Leslie Harvey is a third-year undergraduate student of Computer Engineering at the University of Florida (UF). He works as a research assistant with the Computational Reasoning Group at UF. His research centers around understanding influential factors related to computer science education. As a computer engineering major, he is interested in big data analysis and software engineering.

Ashish Aggarwal, University of Florida

Ashish Aggarwal is an Instructional Assistant Professor of Computer Science in the Department of Engineering Education at the Herbert Wertheim College of Engineering, University of Florida. His research focuses on Computer Science Education and Learning Analytics where he studies the effectiveness of different learning approaches on students' learning outcomes and performance in programming courses.

Exploring the Effect of Quiz and Homework Submission Times on Students' Performance in an Introductory Programming Course in a Flipped Classroom Environment

Abstract

Understanding possible influential factors and their effect on a student's overall performance is imperative for educators to be able to effectively facilitate the comprehension of computer science concepts. With an increase in the amount of research supporting active learning over a traditional learning experience, universities are shifting their course offerings to a flipped classroom environment. This paper focuses on exploring the effect of different students' engagement patterns on their overall course performance in an introductory programming (CS1) course for engineering students in a flipped classroom environment. The course expects students to view prerecorded lecture videos and complete a quiz prior to the scheduled class time, with the in-person instructional period reserved for practicing programming activities. After the class, students are expected to submit a homework assignment based on the concepts learned that week. We are particularly interested in understanding when students submit the quizzes and homework assignments, and if the submission time has any effect on their performance in the course. This paper presents the analysis of 145 students' weekly quiz and homework submission times to understand at which point a student engages with the content and how it affects their performance. Different observed categories are presented to help understand the behavioral aspects of student engagement. Further, analysis is also done considering factors like prior programming experience. We found that students who submitted the quizzes and homeworks 24 hours prior to the submission deadline had significantly higher exam scores as compared to students who submitted during the last 24 hours. Additionally, we also found that this difference was only significant for students who did not have prior programming experience. This indicates that early submission of assignments can help students who do not have prior programming experience in improving their overall course performance. By understanding the major student interaction patterns, we believe that instructors and educators can better design learning experiences for students in CS1 courses.

1 Introduction

Computer programming has historically been difficult for students to learn [1, 2]. As universities begin to admit a higher number of students, the teacher-to-student ratio increases and, thus, strains traditional lecture environments due to the inability to provide more personalized and effective assistance. Consequently, a new learning environment called the "flipped classroom" has emerged with hopes of alleviating this strain on universities. In the flipped classroom model, students benefit from having course instructors readily available to answer questions about in-class activities [3, 4]. This new environment has proven to be effective in STEM fields such as Physics [5], Chemistry [6], and Computer Science [7, 8]. Research shows computer programming education can benefit from adopting the flipped classroom model to teach CS1 courses [4, 9] . While various studies have compared the effectiveness of a flipped class versus a traditional class [9], little is known about the factors which influence students' learning outcome and overall performance. To successfully implement and scale learning through a flipped classroom environment, it is imperative for instructors to know which factors have a dominant effect on students' learning so that they can effectively design their instructions, particularly within their own contexts and coursework [3, 10, 11]. While there are many factors impacting a student's performance [12], this paper presents the study of one such factor, that is quiz and homework submission times, to find if it can influence a student's performance in a CS1 course.

At an R1 research institution in the USA, the CS1 course for engineering students was offered in a flipped classroom. With a new learning environment emerging in many universities, it is important to question how effective this new medium is at cultivating students' conceptual understanding. Furthermore, educational researchers must also investigate how influential different factors are to students in this new learning environment. Therefore, this paper focuses on the quiz and homework submission times and their effect on the overall students' performance. More importantly, quiz time is being investigated since the majority of learning, in this environment, is reliant on a student self-managing their time and engaging with the lecture content outside of the classroom by watching prerecorded lectures. Because course performance is heavily reliant on the student engaging with the course content in the designed manner [6, 13], quiz and homework submission times have a strong potential to indicate their overall behavioral approach towards the course. Here, submission times indicate the time at which students submit an assignment or quiz. This is relatively measured with the difference of the time when the assignment or quiz is due to the time at which the student submits his/her assignment or quiz. Therefore, this paper investigates whether there is statistical support to make a correlation between quiz and homework submission times and average exam performance. Further, prior programming experience is taken into consideration for further analysis which helps in understanding the effect on sub-groups of students present in a class.

2 Background and Related Work

Various researchers have been interested in exploring the influential factors which influence and predict a student's performance in a CS1 course [14, 15]. As the adoption of the flipped classroom pedagogy increases in introductory programming courses, there is a need to explore and analyze the effect of different factors on students' performance in this new flipped format based classes [3, 10, 11]. In this flipped classroom pedagogy, students are expected to familiarize themselves with the weekly course content before the class. This has increased the instructors' expectations from the students as students are now expected to engage with the course content prior to the class. Understanding how outside classroom engagement with the course content affects students' overall course performance can give insights into effective student behavior [14, 16], and can guide the design of flipped CS1 classes. Prior research on analyzing student behavior with their course performance has had mixed results with some suggesting that it is significant [7, 8], and some finding it insignificant [6].

Since students are expected to watch the prerecorded videos before coming to the class, Dazo et al. studied the impact of video viewing behaviors in a flipped CS1 class [13]. They found, firstly, that viewing videos is not guaranteed, and should not be assumed. Secondly, they also found students who watch all the videos earlier, and completely, perform better in the course [13], thus implying the importance of not only watching the videos but also watching them early. Further research indicates students in flipped classrooms are more engaged with their learning and become timelier with their preparation, leading to an increased performance [17, 8]. The flipped classroom model is designed to increase consistent engagement with course content, which minimizes the amount of "cramming" style studying [6]. Participating within a flipped classroom has also been linked to exam performance [6], but the timing of engagement and quality of engagement is largely unknown. Gross et al. indicated that homework accuracy 2 weeks prior can indicate exam performance [6], which suggests time is an influential factor in flipped classrooms. With this in mind, it is evident that understanding student behaviors is necessary to improve the learning in flipped classrooms.

Willman et al., studied the effect of study habits in students' performance in CS1 and found that *"students who receive the highest grade start and finish their work early, do not work on weekends, and do not work at night...."* [18]. Edwards et al., and Shaffer et al., also found that students who begin programming projects earlier perform better than those who begin closer to the due date [16, 19]. Given that it is mostly accepted that students who start earlier on assignments perform better [20], little is known about the effect a student's submission time could potentially have on their performance in a flipped CS1 class.

While many factors can impact a student's learning experience [12, 21], knowing the impact submission time has can be valuable when teachers are incorporating active learning in their courses. Submission time, referred to as the duration between the assignment's due date and the student's submission time, could potentially be more influential in a flipped classroom because of the increased responsibility of the student to guide their learning.

3 Motivation and Research Questions

At an R1 research university in the USA where this research was conducted, the introductory programming course for engineering students was recently offered in a flipped classroom style rather than the traditional lecture based style. The course was divided into multiple sections of 49 students, who met at different times during the week with the instructor. This helped lower the student-to-teacher ratio and facilitated the individualized support mechanism where the instructor and teaching assistants were available to students when they actually worked on the programming problems. While research supports the notion that active learning is more effective in STEM courses [17], it is still unknown what influential factors could enhance or detract a student's learning experience and performance [11].

Because there is an increased responsibility on the student to complete the assigned lectures out of class, it is important to know what kind of impact this has on a student's learning experience [6, 13]. By students managing their own time, some students will submit the required quiz more than a day earlier than some of their peers. With the submission time of quizzes varying between each student, it is important to be able to identify if this impacts a student's overall performance in the course. A starting point for understanding student behaviors is their approach towards self-scheduling the commitments required for a flipped course. Although all students have different schedules impacting when assignments are completed, alongside other factors, this paper strives to understand more about how a student's approach towards the course may impact their performance.

With an understanding of factors influential to a student's overall performance, instructors can convey the results of such findings to students to help them improve their performance. While some students struggle simply due to lack of motivation, some struggle because computer programming is a difficult skill for many to initially grasp [22]. Therefore, this research analyzes patterns in students' quiz submission time and their overall performance by factoring in their prior programming experience. Consequently, our research questions are:

- *How do quiz and homework submission times affect students' performance in an introductory programming course?*
- *What role does prior programming experience have in influencing student's performance with respect to their quiz and homework submission times?*

Knowing more about students' learning behaviors and which type of behaviors contribute to higher performance may lead to reducing the higher failure rate in computer science [4] by altering instructional design which encourages such high-performing behaviors.

4 Course Context

This research paper analyzes the data collected from an introductory programming (CS1) course which teaches programming using MATLAB to engineering students in fall semester of 2019 (before the COVID-19 pandemic). Majority of students enrolled in this course were in their second year of undergraduate studies and were majoring in mechanical, biomedical and aerospace engineering. Students are expected to use this new programming skill in later courses in their undergraduate degree program.

Integral concepts covered are basic syntax, debugging skills, for-loops and while-loops, working with vectors, string manipulation, and image processing along with functions. While this class has historically been taught in a traditional lecture style, where the students attend a weekly two-hour lecture and complete the assigned homework outside of the classroom, it has recently been transformed into a flipped class. By reducing the class size from approximately 300 students enrolled in any given semester into multiple sections of 49, students benefit from the opportunity to have more questions answered, as well as personalized attention.

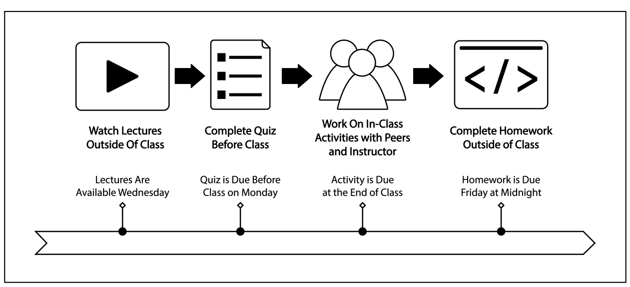

Figure 1: Weekly plan of the CS1 flipped course

Figure 1 shows the weekly plan of the class. Before students come to the weekly class session on Monday, they were expected to have watched all the weekly assigned lecture content and complete a quiz that tests the students on the content they learned. To take advantage of the benefits of active learning, students then worked on in-class programming activities enabling them to utilize the availability of the instructor, teaching assistants, and peers to understand the concepts. After the class, they were assigned a homework assignment, which was due on Friday. Students were required to attend class once a week on Mondays, every week of the semester. Overall, the course was divided into 15 modules, each representing a week in the semester's academic calendar. There were 11 quizzes administered over the entire semester, with 6 quizzes due before the first exam and the remaining 5 due before the second exam. Furthermore, there were 10 homework assignments that students had approximately a week to work on. A total of 145 students were divided into 3 different course sections, at different times of the day on Monday, to facilitate the implementation of the flipped model.

```
clc; clear;
x = 1;
for ii = 1:1:5

    for jj = 1:1:3
        x = x + 3;
    end

    x = x + 2;
end
fprintf('%g', x);
```

Figure 2: Sample quiz question from Module 5

For each module, students were given a week to watch all associated lecture videos and complete the quiz before the start of class. In each module, students are expected to learn most of the content through the lecture videos while supplemental PDF files are additionally provided.

For each module, a quiz based on the module videos was expected to be completed before the beginning of class. Students were given 20 minutes to complete a quiz having 5-10 questions with 2 allowed attempts. Students took the quizzes outside of the classroom wherever and whenever they choose to. Most quiz questions were short responses; however, a few multiple-choice questions were also there. An example of a quiz question from Module 5, covering for-loops, is displayed in Figure 2.

Homework assignments were released to students a week before the due date, typically the Friday following the lecture quiz is due. A homework assignment file consisted of an introduction to the homework topic and use case, lists any prohibited functions, and provided sample test cases to check one's output. If any student needed assistance on an assignment, they could attend office hours or contact the teaching assistants or instructor.

5 Demographics/Chart

Three sections of this course were studied, which had 145 students in total. All of the following information was collected via an introductory survey that each student filled out for a very small portion of their grade. The sections primarily comprised of mechanical engineering (43%), biomedical engineering (20%), aerospace engineering (17%) and other students (20%). A total of 91 students (63%) had no prior programming experience while 54 students (37%) did have prior programming experience . Based on gender, there were 92 (64%) male students, 52 (35%) female students and 2 students (1%) indicated other options. Based on students' academic year, 2 (1%) students were enrolled in first year of their undergraduate program, 120 (83%) students were in their second year, 20 (14%) students were in their third year, 2 (1%) students were in their fourth year and the academic year of 1 (1%) student was unknown.

6 Methodology

To understand how quiz and homework submission time impacts a student's exam performance, data about students' quiz and homework submission times were collected using the learning management system. There were a total of 11 quizzes and 10 homework assignments, including the final project, administered throughout the course. Once the submission time was collected, the submission time was subtracted from the corresponding assignment's due date to produce a result in terms of hours. This value will be referred to as the "time remaining" for each submission. It is important to note that a negative time remaining value indicates a late submission. It is important to note here that this time remaining value has strong potential to characterize a student's behavioral approach towards the course. This value helps in comparative analysis of students based on how much time earlier they submitted the assigned quiz or homework before it was due. While it may be a possibility a student completes a particular assignment well ahead of the deadline and does not submit it until the last minute when it is due, a collection of multiple such time remaining values can still indicate a dominant behavioral approach, especially when analyzing it over a semester's coursework. Thus, these time remaining values form the bases of our characterization of a student's approach towards a course, which we believe to be valuable in examining for its potential impact on students' performance in a flipped class environment.

Once the time remaining value of all the 11 quizzes was collected for each student, a median quiz time remaining value for each student was produced. Median was used in this instance, instead of average, to ensure outliers did not skew the categorization of the overall quiz time remaining of a student significantly. Similarly, the median homework time remaining value of a student was calculated by using the time remaining values for all 10 homework submissions. To produce a single time remaining value for each student, the median of the median quiz time remaining value and the median homework time remaining value was calculated for each student. This value is referred to as the "aggregate time remaining" value. In this way, a single data point was produced that aided in the categorization of a particular student's aggregate time remaining value. Median, instead of average, was again used since using the average significantly altered the results because of outliers. Through this, the approach a student took towards assignments was quantified and used in clustering the students in different categories.

Categories were created based on major clusters formed when analyzing the students' aggregate time remaining values. These categories are: '$\leq$ 6 hours', '$>$ 6 hours and $\leq$ 12 hours', '$>$ 12 hours and $\leq$ 24 hours', and '$>$ 24 hours'.

Student's prior programming experience was collected via an introductory survey administered at the beginning of the course. The question about the student's prior programming experience (PPE) had the following options to choose from: "No prior experience", "Between 1 to 10 hours", "Between 11 to 100 hours", "Between 101 to 500 hours", "I am a software developer", or "I invented a programming language". After choosing a response indicating prior programming experience, the student was asked for a description of their experience. If a student was determined to have a basic understanding of programming, they were assigned a value of 1. All students deemed having no PPE or very little experience were assigned a value of 0. It is important to note students' self-reported prior programming experience and description of their experience were taken into account when assigning these binary PPE values.

The course was comprised of 2 exams: a midterm and a final. Each exam consisted of 6 questions in total for 100 points. The first question (20 points) had 4 different code snippets where students were expected to predict the output of the programs. The second question (10 points) was on debugging, where students were expected to identify and correct the errors in a given program. The third and the fourth question (15 points each) were short programming problems, and the fifth and sixth questions (20 points each) were more comprehensive programming problems where students were expected to write the programs. The average of these scores created a value that will be referred to as "exam score" throughout this paper for analysis purposes. While this may not give a complete measure of a student's performance, it was used to gain an overall understanding of a student's learning.

Firstly, we analyzed to see if different class sections conducted during different times have an impact on the overall student's submission times. No statistical differences in the submission times of the students based on the class timings were observed. This indicates that we can aggregate all of the 145 students into one dataset for further analysis.

We are interested in knowing if there is a mean difference on exam score between students whose aggregate time remaining is $\leq$ 6 hours (Group 1), $>$ 6 hours and $\leq$ 12 hours (Group 2), $>$ 12 hours and $\leq$ 24 hours (Group 3), $>$24 hours (Group 4). Table 1 shows the group numbers and their corresponding "aggregate time remaining" values.

The null hypothesis is that there is no difference in the exam score means of the 4 groups, and the alternate hypothesis is that there is a difference in the exam score means of the 4 groups. There are 145 students classified according to their aggregate time remaining values into the 4 groups (Table 1). Figure 3 shows the number of students present in every group. For Group 1 (n = 26), for Group 2 (n = 37), for Group 3 (n = 39), and for Group 4 (n = 43). All students took both exams, midterm and final, and the average of both exams is calculated and referred to as exam score for each student. For the analysis, the submission time groups was the independent variable and the exam score was the outcome variable. A one-way ANOVA was conducted to determine if the mean of the exam score differed in the 4 groups. The assumption of normality was tested and met via examination of the residuals. According to Levene's test, the homogeneity of variance assumption was satisfied [F (3, 141) = 0.691, p = .559].

Group	Aggregate Time Remaining
1	6 hours
2	6 hours and $\leq$ 12 hours
3	12 hours and $\leq$ 24 hours
4	24 hours

Table 1: Group categorization based on submission times

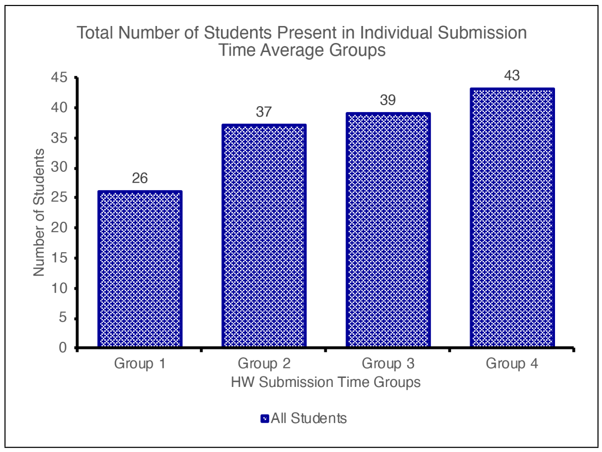

Figure 3: Total number of students in each of the 4 groups

7 Findings

We find that one-way ANOVA is statistically significant (F = 3.280, df = 3, 141, p = 0.023). The means (Figure 4) and the standard deviations of the exam average score for each group of the independent variable were as follows: 81.03 (SD = 10.21) for Group 1, 78.49 (SD = 12.30) for Group 2, 81.25 (SD = 11.21) for Group 3, and 85.98 (SD = 9.92) for Group 4. The means in Figure 4 show students in Group 4 (aggregate time remaining $>$ 24 hours) have higher exam scores than students in other groups. A Bonferroni test was conducted (alpha = 0.05) which confirmed there is a statistical difference (p = 0.017) between individuals in Group 2 (aggregate time remaining $>$ 6 hours and $\leq$ 12 hours) and Group 4 (aggregate time remaining $>$ 24 hours). However, there is no statistical difference between students in other combinations of groups.

To further understand this significant difference and the source of this significant difference, the next analysis is performed by the differentiation of the students based on prior programming experience (PPE). In the past, many studies have indicated the role of PPE in students' performance in CS1 courses [23]. Therefore, it was an important factor to consider when further exploring the source of difference found in the above results. Students' separate one-way ANOVA was conducted on individuals having PPE = 0 (students with no prior programming experience) and PPE = 1 (students with prior programming experience).

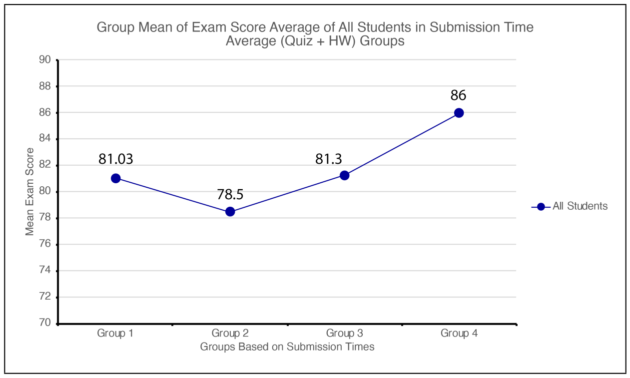

Figure 4: Mean exam score of all the students present in respective groups

7.1 Results from the Analysis of Students with PPE = 1

There were a total of 54 students (Figure 5) having prior programming experience who were classified according to their aggregate time remaining using the prior defined groups, Group 1: n = 8, Group 2: n = 16, Group 3: n = 15, and Group 4: n = 15. For the analysis, the submission time group category was the independent variable and the exam score was the outcome variable. A one-way ANOVA was conducted to determine if the mean of the exam score differed in the 4 categories. The assumption of normality was tested and met via examination of the residuals. According to Levene's test, the homogeneity of variance assumption was satisfied [F (3, 50) = 0.725, p = .543]. ***The one-way ANOVA is <u>not</u> statistically significant (F = 0.652, df = 3, p = 0.585)***. The means and the standard deviations of the exam average score for each group of the independent variable were as follows: 86.15 (SD = 5.19) for Group 1, 82 (SD = 10.05) for Group 2, 85.38 (SD = 8.75) for Group 3, and 85.70 (SD = 9.18) for Group 4. The means and the one-way ANOVA result indicate that group classification is not an important factor in exam score. While the differences in the exam score are statistically significant for all the students (n = 145), when considering students with prior programming experience it is not found to be statistically significant. This calls for analyzing students with no prior programming experience using similar statistical analysis. The following section presents the results when analyzing students with no prior programming experience.

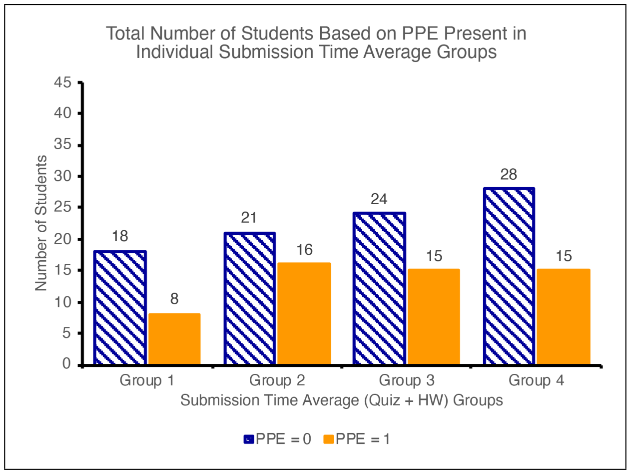

Figure 5: Total number of students present in individual groups based on PPE

7.2 Results from the Analysis of Students with PPE = 0

There were a total of 91 students (Figure 5) having no prior programming experience who were classified according to their aggregate time remaining using the prior defined groups, Group 1: n = 18, Group 2: n = 21, Group 3: n = 24, and Group 4: n = 28. For the analysis, the submission time group category was the independent variable and the exam score was the outcome variable. A one-way ANOVA was conducted to determine if the mean of the exam score differed in the 4 categories. The assumption of normality was tested and met via examination of the residuals. According to Levene's test, the homogeneity of variance assumption was satisfied [F (3, 87) = 0.765, p = .517]. ***The one-way ANOVA is statistically significant (F = 0.3.605, df = 3, p = 0.017)***. The means and the standard deviations of the exam average score for each group of the independent variable were as follows: 78.75 (SD = 11.16) for Group 1, 75.80 (SD = 13.38 for Group 2, 78.67 (SD = 11.95) for Group 3, and 86.13 (SD = 10.45) for Group 4. A Bonferroni test was conducted (alpha = 0.05) which confirmed that there is a statistical difference (p = 0.18) between individuals in Group 2 (aggregate time remaining $>$ 6 hours and $\leq$ 12 hours) and Group 4 (aggregate time remaining $>$ 24 hours).

8 Discussion

From the above discussed findings, it was observed that, overall, there is a statistical difference in exam score means of the 4 groups. However, on further analysis, it is observed that the source of this difference comes from students having no prior programming experience. No statistical difference is found within students having prior programming experience, yet it was discovered when analyzing students with no prior programming experience. This indicates that the quiz and homework assignment submission times impact the exam performance only for students with no prior programming experience. Below is a line plot (Figure 6) of the exam score means for students in each group. The two lines represent students with prior programming experience and students without prior programming experience.

This line plot indicates 2 important observations. First, it is observable that students with PPE = 0 have a lower exam score mean than students with PPE = 1 for most of the groups. The second observation is the mean exam score of the students in Group 4 is similar regardless of their prior programming experience. This suggests that not only the early homework and quiz submission times of a student have a positive impact on their exam performance, but students without prior programming experience can overcome the disadvantage associated with having no prior programming experience by early submissions of quiz and homework assignments. While many factors contribute to a student's performance in a course, we believe students in Group 4 performed better since there is more time available for a student's comprehension of the concepts while solving a programming problem. This enables the student to understand assignments at a deeper level rather than simply completing them at the last minute. This is also supported by observing how students in Group 4, with no prior programming experience, perform similarly to students who have prior programming experience, which is not the case in other groups.

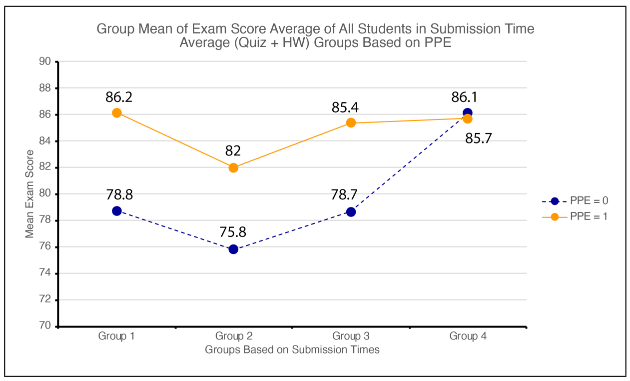

Figure 6: Mean of exam score based on PPE in respective groups

These findings are consistent with the prior research findings of Edwards et al., [16], Shaffer et al., [19] and Willman et al., [18] which found that students perform well in CS1 course if they start and submit their assignments early. These findings also resonate with the findings of Dazo et al., [13] which suggests that students who watch videos earlier tend to perform better. These findings are consistent with this because an early quiz submission time does indicate early video watching, and overall approaching the course assignments earlier. The findings are also consistent with what Gross et al., [6] suggest in their research, that preclass preparation improves students' outcomes in a flipped class.

These findings have the potential to impact our understanding of what contributes to a student's success in a flipped CS1 course. While there have been many studies that have found the effectiveness of different factors in a student's success, we believe that having a contextual factor analysis can help guide the instruction of the course. As growing number of students with mixed abilities and different backgrounds enroll in CS1, we believe these findings will enable instructors to better scaffold students' learning by encouraging certain types of behaviors either through instructional design or explicit recommendation. Based on our findings, we suggest instructors should find ways to encourage students with no prior programming experience to begin assignments early. Instructors relaying this information to students with no prior programming experience may decrease initial concerns from students skeptical about performing at the same level as students with prior programming experience.

9 Limitations

One of the limitations of the study was that an early quiz submission does not indicate more time spent digesting the material. Students submitting assignments closer to the due date may be more motivated to complete the assignment for credit, while students submitting earlier may have completed the assignment to understand it more. This is a large assumption given that various factors influence when a student will complete an assignment. These could range from schedule conflicts with other classes to more personal obstacles. Therefore, additional surveys would need to be administered to provide more insight into this assumption.

10 Future Work

As many contributing factors influence a student's exam performance, analyzing quiz and homework submission time is only one of many factors. We believe the next factors worth exploring are the perception and motivation a student has about the course. Additionally, it would be interesting to compare the perception of students before and after the course, how that perception of programming changes over the course, and how it influences their overall performance. Another avenue to explore would be to collect more data on students' learning behavior and build predictive and explainable models which can help in understanding aggregate learning behavioral patterns and their impact on learning outcomes.

11 Conclusion

Through analyzing student's aggregate homework and quiz submission times, we were interested in exploring the impact these variables play on exam performance. By performing different statistical tests, it was observed students with prior programming experience are not impacted by their aggregate submission time. However, students with no prior programming experience can benefit from submitting their quiz and homework assignments earlier, which increases their mean exam performance. The results of this study imply the importance of students' engagement patterns on their performance based on their quiz and homework submission time. With respect to the implementation of the flipped class model, the results indicate that there are certain types of engagement patterns which may be more rewarding than others, especially for certain groups of students. As the use of the flipped classroom model increases, it is valuable for instructors and students to know constructive behavior and engagement patterns that have a positive impact on students' performance, in the context of individual courses, so that such behavior is encouraged and recommended.

12 Acknowledgment

We would like to thank the anonymous reviewers for their valuable feedback and suggestions.

References

[1] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," *SIGCSE Bull.*, vol. 39, no. 2, p. 32–36, Jun. 2007. [Online]. Available: https://doi.org/10.1145/1272848.1272879

[2] C. Watson and F. W. Li, "Failure rates in introductory programming revisited," in *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science*

Education, ser. ITiCSE '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 39–44. [Online]. Available: https://doi.org/10.1145/2591708.2591749

[3] J. Bishop and M. A. Verleger, "The flipped classroom: A survey of the research," in *2013 ASEE Annual Conference & Exposition*, no. 9. Atlanta, Georgia: ASEE Conferences, June 2013, pp. 1–18. [Online]. Available: https://peer.asee.org/22585

[4] C. Latulipe, A. Rorrer, and B. Long, "Longitudinal data on flipped class effects on performance in cs1 and retention after cs1," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 411–416. [Online]. Available: https://doi.org/10.1145/3159450.3159518

[5] L. Deslauriers, L. S. McCarty, K. Miller, K. Callaghan, and G. Kestin, "Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom," *Proceedings of the National Academy of Sciences*, vol. 116, no. 39, pp. 19 251–19 257, 2019. [Online]. Available: https://www.pnas.org/content/116/39/19251

[6] D. Gross, E. Pietri, G. Anderson, K. Moyano Camihort, and M. Graham, "Increased pre-class preparation underlies student outcome improvement in the flipped classroom," *CBE life sciences education*, vol. 14, 12 2015.

[7] A. Herala, E. Vanhala, A. Knutas, and J. Ikonen, "Teaching programming with flipped classroom method: A study from two programming courses," in *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, ser. Koli Calling '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 165–166. [Online]. Available: https://doi.org/10.1145/2828959.2828983

[8] R. Caceffo, G. Gama, and R. Azevedo, "Exploring active learning approaches to computer science classes," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 922–927. [Online]. Available: https://doi.org/10.1145/3159450.3159585

[9] D. Horton, M. Craig, J. Campbell, P. Gries, and D. Zingaro, "Comparing outcomes in inverted and traditional cs1," in *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ser. ITiCSE '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 261–266. [Online]. Available: https://doi.org/10.1145/2591708.2591752

[10] M. N. Giannakos, J. Krogstie, and N. Chrisochoides, "Reviewing the flipped classroom research: Reflections for computer science education," in *Proceedings of the Computer Science Education Research Conference*, ser. CSERC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 23–29. [Online]. Available: https://doi.org/10.1145/2691352.2691354

[11] A. Karabulut-Ilgu, N. Jaramillo Cherrez, and C. T. Jahren, "A systematic review of research on the flipped learning method in engineering education," *British Journal*

of Educational Technology, vol. 49, no. 3, pp. 398–411, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/bjet.12548

[12] B. C. Wilson and S. Shrock, “Contributing to success in an introductory computer science course: A study of twelve factors,” *SIGCSE Bull.*, vol. 33, no. 1, p. 184–188, Feb. 2001. [Online]. Available: https://doi.org/10.1145/366413.364581

[13] S. L. Dazo, N. R. Stepanek, R. Fulkerson, and B. Dorn, “An empirical analysis of video viewing behaviors in flipped cs1 courses,” in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 106–111. [Online]. Available: https://doi.org/10.1145/2899415.2899468

[14] S. N. Liao, S. Valstar, K. Thai, C. Alvarado, D. Zingaro, W. G. Griswold, and L. Porter, “Behaviors of higher and lower performing students in cs1,” in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 196–202. [Online]. Available: https://doi.org/10.1145/3304221.3319740

[15] K. Mierle, K. Laven, S. Roweis, and G. Wilson, “Mining student cvs repositories for performance indicators,” *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, p. 1–5, May 2005. [Online]. Available: https://doi.org/10.1145/1082983.1083150

[16] S. H. Edwards, J. Snyder, M. A. Pérez-Quiñones, A. Allevato, D. Kim, and B. Tretola, “Comparing effective and ineffective behaviors of student programmers,” in *Proceedings of the Fifth International Workshop on Computing Education Research Workshop*, ser. ICER ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 3–14. [Online]. Available: https://doi.org/10.1145/1584322.1584325

[17] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, “Active learning increases student performance in science, engineering, and mathematics,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014. [Online]. Available: https://www.pnas.org/content/111/23/8410

[18] S. Willman, R. Lindén, E. Kaila, T. Rajala, M.-J. Laakso, and T. Salakoski, “On study habits on an introductory course on programming,” *Computer Science Education*, vol. 25, no. 3, pp. 276–291, 2015.

[19] C. A. Shaffer and S. H. Edwards, “Scheduling and student performance,” in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 331. [Online]. Available: https://doi.org/10.1145/1999747.1999842

[20] J. Spacco, P. Denny, B. Richards, D. Babcock, D. Hovemeyer, J. Moscola, and R. Duvall, “Analyzing student work patterns using programming exercise data,” in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 18–23. [Online]. Available: https://doi.org/10.1145/2676723.2677297

[21] M. Säde, R. Suviste, P. Luik, E. Tõnisson, and M. Lepp, "Factors that influence students' motivation and perception of studying computer science," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 873–878. [Online]. Available: https://doi.org/10.1145/3287324.3287395

[22] P. Kinnunen and L. Malmi, "Why students drop out cs1 course?" in *Proceedings of the Second International Workshop on Computing Education Research*, ser. ICER '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 97–108. [Online]. Available: https://doi.org/10.1145/1151588.1151604

[23] C. Wilcox and A. Lionelle, "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*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 80–85. [Online]. Available: https://doi.org/10.1145/3159450.3159480