ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #37417

Work-in-Progress: Relationship of Students' Class Preparation and Learning in a Flipped Computer Programming Course

Kwansun Cho (Lecturer)

Kwansun Cho is an Instructional Assistant Professor of the Department of Engineering Education, in the UF Herbert Wertheim College of Engineering. She has been teaching introductory computer programming courses for engineers. She holds two Masters' degrees in Electrical and Computer Engineering from the University of Florida and Yonsei University, specializing in speech signal processing. Her educational research interests include improved flipped classroom teaching/learning for students, and computer- or web-assisted personalized learning.

Saira Anwar

Saira Anwar is an Assistant Professor at the Department of Multidisciplinary Engineering, Texas A&M University, College Station. She received her Ph.D. in Engineering Education from the School of Engineering Education, Purdue University, USA. She earned her M.S. in Computer Science with a software engineering concentration from the National University of Computer and Emerging Sciences, Pakistan. Dr. Anwar also holds an M.Sc in Computer Science from Punjab University College of Information Technology, Pakistan. Dr. Anwar is passionate about research and teaching, specifically translating research into evidence-based teaching practices. For her research, she is particularly interested in designing interventions that help develop students' understanding of conceptually hard concepts in STEM courses. She was awarded the 2020 outstanding researcher award by the School of Engineering Education, Purdue University. Also, she is the recipient of Apprentice Faculty Grant Award, 2022 by ERM division, ASEE. Dr. Anwar has over 13 years of teaching experience, including the University of Florida (Department of Engineering Education), Forman Christian College University (Department of Computer Science), and many other higher education institutes in Pakistan. She taught engineering education, computer science, and software engineering courses. She believes in implementing engaging, motivating, and interactive learning experiences through curricular innovation. She was awarded outstanding teacher awards in 2013 and 2006 by her then employers. Also, she was the recipient of the "President of Pakistan Merit and Talent Scholarship" for her undergraduate studies.

© American Society for Engineering Education, 2022
Powered by www.slayte.com

Work-in-Progress: Relationship of Students' Class Preparation and Learning in a Flipped Computer Programming Course

Abstract

This work-in-progress paper examines the relationship between students' preparation in flipped class and their learning.

In engineering education, flipped course design is getting more attention among instructors of STEM courses. Flipped classroom model emphasizes student-centered learning, where much of students' learning is connected to their preparation before coming to the class using videos and other study material. As the model is highly dependent on students' self-preparation, it is crucial to capture the trends of students' preparation and its impact on students' learning for effective course design and continuous improvement.

This study presents the preliminary results of an online flipped C++ programming course and evaluates the relationship between undergraduate students' class preparation and learning. Data were collected from 66 students for the whole semester, comprising 15 weeks. For preparation, students were encouraged to watch two videos for the flipped class: 1) description of programming construct and concept and 2) instructor emulation of a live coding session. For measuring students' class preparation, we recorded the video analytics indicating the time spent by each student to watch both videos respectively in each week. In addition, we used students' final scores in the course to measure students' learning and evaluated the relationship between students' class preparation and learning. Furthermore, we examined the trends of time spent on video watching for each week.

Preliminary analysis was conducted using multiple regression and repeated measures ANOVA. The results indicate a significant relationship between students' preparation (time spent on videos) and their learning (final score). Further, the trends in repeated measures highlight the weeks where students spent the most time preparing. This work-in-progress paper relates the study results with the course design.

Introduction

In engineering education, flipped course design is getting more attention among instructors of STEM courses [1]. Prior studies suggest that flipped courses were designed and evaluated for effectiveness and factors influencing students’ learning in computer science and information technology disciplines [2], [3]. For example, Amresh and colleagues [2] compared students' scores in two flipped class sections with a traditional section for an introductory programming course. The authors reported that the flipped class approach positively impacted students' performance.

Existing studies in programming courses suggest that flipped class approach can be valuable because it provides more opportunities and time for students to interact with their instructor and peers. Additionally, students can get timely help and assistance from instructors and peers while they engage in-class hands-on activities and assignments during class hours [4]. However, for flipped courses, student success is highly dependent on their self-preparation using provided study material, including videos, lecture notes or slides, and book chapters before coming to class [5], [6], which could be challenging for novice programmers [7]. Diwanji and colleagues [6] highlighted the importance of students' preparation in a flipped classroom model. They conducted a student survey and found that only a few students (less than 30%) in the flipped class prepared before the class. Their analysis suggested Artificial Intelligence-based applications and technology tools to boost students' extrinsic and intrinsic motivation for preparation. Also, the authors emphasized the importance of evaluating students' class preparation for flipped classes. Considering the importance of students' preparation in flipped classes for their learning, in this work-in-progress paper, we examined the relationship between students' preparation and performance. More specifically, the paper addresses the following research questions.

1. How does students' class preparation relate to their learning in a flipped programming course?
2. What are the trends in students' class preparation with respect to the programming constructs?

Literature Review

Prior studies have found student-centered learning as more effective than the traditional model of instruction [8], [9]. Student-centered learning can be defined as learning approaches that place students at the center of their learning process. These approaches influence students' learning through the content, activities, materials, and place of learning [10]. A broad range of models that implement student-centered learning has been developed, including active learning, collaborative learning, inquiry-based learning, cooperative learning, problem-based learning, team-based learning, peer instruction, small group learning, and project-based learning [10].

Previously, these student-centered learning models have been used in computer science and engineering courses, specifically courses with conceptually hard concepts such as programming courses. These models include but not limited to active learning [e.g., 11], collaborative learning [e.g., 12], cooperative learning [e.g., 13], flipped classroom learning [e.g., 14], inquiry-based learning [e.g., 15], problem based learning [e.g., 16], and project-based learning [e.g., 17].

Flipped classroom model is one of the widely accepted models for enhancing students' knowledge using student-centered learning [18]. In the recent pandemic era, many institutes around the U.S. started to use flipped classroom model, which is extensively adapted, especially for delivering programming courses. The main idea for flipped classroom model is to let students learn basic course content outside the class individually by watching pre-recorded lectures or videos and/or reading textbooks. Further, during in-class hours, students apply their learned content knowledge on application through hands-on activities and active learning techniques such as problem-based learning and peer instruction with the help of instructors and peers during the class hours [19], [20].

Previous literature has examined the role of the flipped class on students' learning or performance by comparing students' scores in the flipped class with those in the traditional class or by collecting students' perceptions with surveys or interviews [2], [14]. For example, Pattanaphanchai [14] investigated students' learning achievement and perception in an introductory programming course using a flipped class. The author reported students' satisfaction with flipped class and reported that learning in a flipped class with in-class practice and out-of-class studying helped them to understand programming concepts. Also, the authors reported that the students in the flipped class environment significantly outperformed those in a traditional lecture class.

Although extensive literature has compared flipped classroom model with traditional classes, there is little work investigating students' learning or performance within flipped classes by measuring students' self-preparation time spent with study materials such as videos, which is the premise of this work-in-progress paper.

Research Methods

Site and participants

For this work-in-progress paper, we designed a cross-sectional study and collected the data from 66 undergraduate engineering students enrolled in a large R1 South-Eastern university. The data were collected from two sections (taught by the same instructor) in Spring 2021. The students were enrolled in a C++ programming course titled "C++ programming for engineers." The course was an introductory course for students with little or no programming experience and provided them with a hands-on learning experience in the C++ programming language. The course focused on developing students' problem-solving and computational thinking abilities by devising software solutions for various engineering problems. For providing students an experience with real problems during the class, the course utilized flipped classroom model. The class was divided into three different phases 1) Before the class, 2) During the class, and 3) After the class. Before the class, students were expected to watch two videos and read study material before attending the week's classes (except week 1) throughout the semester (15 weeks long). During the class, students were required to code their algorithms in C++ for solving three simple yet relevant in-class assignments, focusing on enhancing their problem-solving abilities and devising solutions using computational thinking. Students were required to submit their complete work by the end of the class session to get full credit for the weekly in-class assignments. While students worked on these in-class assignments, the instructional team (instructor and peer mentors (senior students hired as teaching assistants who had demonstrated excellence when taking the course)) acted as facilitators and scaffolded students to help in their learning process by providing personalized feedback. The course instructor graded these assignments. In Spring 2021, the class was conducted using an online synchronous session with required attendance. After the class, students were asked to solve more challenging homework problems related to the same content topic they watched before class and solved programming problems during class time. Students were required to solve these problems individually, where peer mentors and instructors conducted office hours to help students during their logic building and problem-solving process.

Measures

From both sections of the course, we collected the data on students' preparation for the class and students' learning. For students' preparation for the class, as associated with students' watching the videos, we collected the data through class videos created to introduce students to programming concepts. The instructor made two videos for each week (except week 1). The first video described the programming construct and concepts needed for that week (Concept video). The video emphasized giving students all information they need to know the constructs, including syntax and semantics. And, it demonstrated that how the construct could be used as a building block in the larger scheme of the program development. The second video was designed to enhance students' knowledge by the instructor emulating the live coding session for the same concepts and constructs (Coding video). The instructor used a set of sample problems and solved them for students step-by-step from design to execution. For each student, each week, we used video analytics, indicating the time spent by each student watching both the videos. For students' learning, we used students' final scores. The final score out of 100 points was the cumulative total indicative of students' performance from in-class assignments, homework assignments, and three exams.

Data Analysis

To answer the two questions of this work-in-progress paper, we conducted the preliminary analysis using multiple regression analysis and repeated measures ANOVA. For analysis, we used SPSS 28.0 and examined the descriptive statistics to test both methods' assumptions. We examined the outliers, measured skewness, and kurtosis, and tested data for normality and homogeneity. In addition, we explored issues of multicollinearity using the variance inflation factor and found no issues. Further, we tested the sphericity using Mauchly's W test. In case of sphericity violation, we examined the epsilons for adjusting the degrees of freedom. We used 0.75 as the cutoff value and used Huynh-Feldt epsilons for adjustments if the value was greater than 0.75 or Greenhouse-Geisser otherwise.

For regression analysis, we calculated the total time in seconds that students spent watching video1 (Concept video) and video2 (Coding video).

Preliminary Results

We used multiple regression analysis for the first research question to examine the relationship between students' class preparation and learning. We used the total time of both concept and coding videos as independent variables and used students' total scores as the dependent variable. The results of the analysis are presented in Table 1. We are depicting results using standardized coefficients.

Table 1. Regression analysis between students' preparation (time in seconds) and learning

Estimate	B	SE	*t*	*P*	sr^2
Constant	87.272	1.523	57.287	<.001**	
Concept video	.427	.000	2.632	.011*	.332
Coding video	-.524	.000	-3.229	.002**	-.407

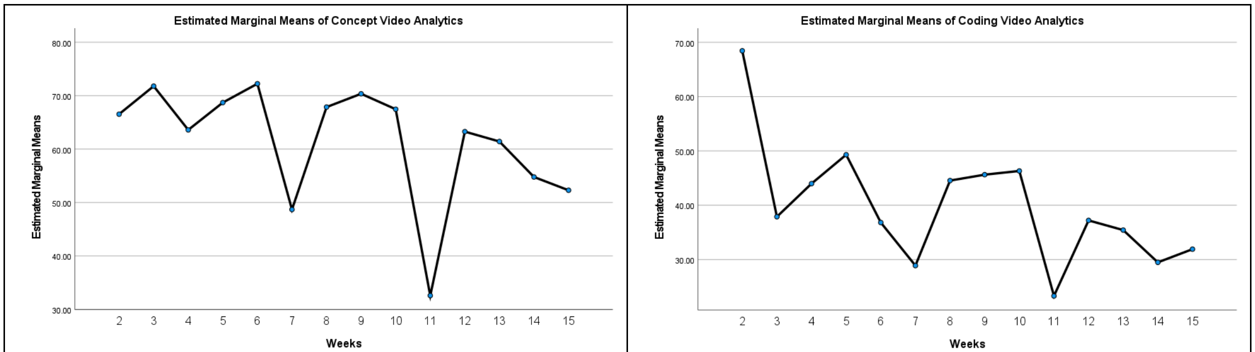
** p<0.05, **p<.01*

The results indicate a significant relationship with $F(2,52) = 5.508$, $p = .007$ between students' preparation (measured using video analytics of watching time) and students learning (i.e., the total score in the course). Additionally, $R^2 = .175$ indicates that measures of students' preparation account for a 17.5% variance in students' learning. Further, both concept videos and coding videos were significant contributors to students' learning, with one standardized unit increase in the watching time of concept videos increasing students' learning by .427 units. However, the same was not valid for coding videos, and it had an inverse relationship with students' learning. We used repeated-measures ANOVA to explore students' class preparation trends with programming constructs for the second research question. Table 2 presents the results of repeated measures ANOVA.

Table 2. Results of the repeated measures ANOVA

	W	ε	χ^2	F	p	η^2
Concept video	.010	.561	276.599	$F(7.288, 473.744) = 6.256$	<.001**	.088
Coding video	.023	.627	228.087	$F(8.156, 530.161) = 7.971$	<.001**	.109

*$p<0.05$, **$p<.01$



A) Concept video B) Coding video

Figure 1. Trends of time spent in watching the videos across weeks

The result indicates a significant mean difference of time in 15 weeks of the whole semester (except week 1). Figure 1 (A&B) shows the trends of students' preparation in concept and coding videos, respectively, across the 15 weeks (except week 1). Also, the Bonferroni post-hoc analysis indicates that for Concept video, the most significant declines were observed at week 7 and week 11. Both week 3 (MD= 23.091) and week 6 (MD =23.545) had a significant mean difference from week 7. Also, week 2 (MD=33.954), week 3 (MD=39.200), week 4 (MD=36.128), week 5 (MD=36.128), week 6 (MD=39.654), week 8 (MD=35.283), week 9 (MD=37.773), week 10 (MD=34.880), week 12 (MD=30.689), and week 13 (MD=28.841) reported significant mean difference with week 11.

For the coding video, we observe a declining trend from week 2 to week 15. This is also evident from the Bonferroni post hoc analysis where week 3 (MD=30.551), week 4 (MD=24.451), week 6 (MD=31.614), week 7 (MD=39.545), week 8 (MD=23.889), week 11 (MD=45.137), week 12 (MD=31.225), week 13 (MD=33.004), week 14 (MD=38.947), and week 15 (MD=36.520) show a significant mean difference from week 2. Additionally, week 4 (MD= 20.685), week 5 (MD=20.411), and week 10 (MD=23.017) have a significant mean difference from week 11. Also, week 5 has a significant mean difference from week 7 (MD=20.411).

Discussion and Conclusion

Programming is conceptually challenging for students, especially novice students [21] and in the flipped classroom model, students' learning depends on their preparation [5], [6]. This work-in-progress paper investigated students' preparation relationship with their C++ programming course learning. The paper's findings are interesting as for both concept videos and coding videos, weeks 7 and 11 showed a significant decline. Once correlated with the course syllabus and weekly topics, we found that in week 7, the arrays conceptualizations were discussed, while in week 11, students studied strings in C++. Considering that strings are character arrays, these topics have higher overlap. In the full paper, it is essential to evaluate the conceptual difficulty of these concepts and what makes them less interesting for students. Also, the full paper may explore the videos' content to explore besides conceptual construct, what aspects of videos are making the difference in students' watching time and their learning. Also, it will be interesting to triangulate the results with coding videos to understand the declining trend and why these videos have an inverse relationship with students' learning. It is noteworthy that students' started the semester with relatively higher watching time for both videos. However, for concept videos, they watched the videos for more time in week 3 (branching and looping), week 6 (recursion), week 8 (structures and classes), week 9 (constructors), week 10 (operator overloading), and week 12 (pointers). However, they relatively spent more time on coding videos in week 5 (parameters and overloading), week 8 (structures and classes), week 9 (constructors), and week 10 (operator overloading). Both videos were week 8, week 9, and week 10, which are associated with object-oriented programming in C++, indicating students either found them difficult [22] or worth watching. Future studies and full papers may view these results with the conceptual difficulty of the construct and students' past experiences with programming.

The results of this paper may be viewed with several limitations. These include the relatively small sample size. Future studies may use longitudinal research design to examine the trends across semesters, multiple years, and other programming courses by collecting and analyzing data with a larger sample size. Additionally, the study has limitations associated with correlational studies, which future studies may overcome. Due to the smaller sample size, we couldn't account for race and gender-based variations. Future studies with more data may also account for these variations to provide insights from marginalized groups' perspectives. Also, we used only one measure of students' preparation without any classroom observation data using structured on unstructured protocols [23], student engagement data from in-class videos, students' perceptions, perceived needs before coming to the class [24], or the instructor's perspective of students' preparation. Future studies may counter for the same. Future studies may also investigate students' learning or performance in a hybrid model that combines traditional classroom and flipped classroom models for computer programming courses, especially as prior studies have highlighted the challenging nature of flipped classroom model for novice programmers [7], [21]. Further, future studies may employ the novel methodologies to understand the impact of such strategies on students' learning in real-time, using other noncognitive markers such as eye-tracking and students' emotions while watching the videos [25].

References

[1] A. Karabulut-Ilgu, N. Cherrez, and C. 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, Feb. 2017.
[2] A. Amresh, A. Carberry, and J. Femiani, "Evaluating the effectiveness of flipped classrooms for teaching CS1," in *IEEE Frontiers in Education Conference, FIE 2013, Oklahoma City, OK, USA, Oct. 23-26, 2013,* pp. 733-735.
[3] S. Mithun and X. Luo, "Design and evaluate the factors for flipped classrooms for data management courses," in *IEEE Frontiers in Education Conference, FIE 2020, Uppsala, Sweden, Oct. 21-24, 2020,* pp. 1-8.
[4] J. Sun and Y. Wu, "Analysis of learning achievement and teacher-student interactions in flipped and conventional classrooms," *The International Review of Research in Open and Distributed Learning*, vol. 17, no. 1, pp. 79–99, Jan. 2016.
[5] D. Gross, E. Pietri, G. Anderson, K. Moyano-Camihort, and M. Graham, "Increased preclass preparation underlies student outcome improvement in the flipped classroom," *CBE-Life Sciences Education*, vol. 14, no. 4, article 36, Dec. 2015.
[6] P. Diwanji, K. Hinkelmann, and H. Witschel, "Enhance classroom preparation for flipped classroom using AI and analytics," in *International Conference on Enterprise Information Systems, ICEIS 2018, Funchal, Madeira, Portugal, Mar. 21-24, 2018,* pp. 477-483.
[7] S. Alhazbi and O. Halabi, "Flipping introductory programming class: Potentials, challenges, and research gaps," in *International Conference on Education Technology and Computers, ICETC 2018, Tokyo, Japan, Oct. 26-28, 2018,* pp. 27–32.
[8] G. Catalano and K. Catalano, "Transformation: From teacher-centered to student-centered engineering education," *Journal of Engineering Education*, vol. 88, no. 1, pp. 59-64, Jan. 1999.
[9] S. Anwar and M. Menekse, "Unique contributions of individual reflections and teamwork on engineering students' academic performance and achievement goals," *International Journal of Engineering Education*, vol. 36, no. 3, pp. 1018-1033, May 2020.
[10] J. Froyd and N. Simpson, "Student-centered learning addressing faculty questions about student-centered learning," in *Course, Curriculum, Labour, and Improvement Conference, Washington DC, USA, Aug. 2008,* vol. 30, no. 11, pp. 1-11.
[11] J. Berssanette and A. de Francisco, "Active learning in the context of the teaching/learning of computer programming: a systematic review," *Journal of Information Technology Education: Research*, vol. 20, pp. 201-220, 2021.
[12] Y. Hayashi, K. Fukamachi, and H. Komatsugawa, "Collaborative learning in computer programming courses that adopted the flipped classroom," in *IEEE International Conference on Learning and Teaching in Computing and Engineering, LaTiCE 2015, Taipei, Taiwan, Apr. 9-12, 2015,* pp. 209-212.
[13] M. Garcia, "Cooperative learning in computer programming: A quasi-experimental evaluation of Jigsaw teaching strategy with novice programmers," *Education and Information Technologies*, vol. 26, no. 4, pp. 4839-4856, July 2021.
[14] J. Pattanaphanchai, "An investigation of students' learning achievement and perception using flipped classroom in an introductory programming course: A case study of Thailand higher education," *Journal of University Teaching and Learning Practice*, vol. 16, no. 5, article 4, 2019.

[15] S. Coleman and E. Nichols, "Embedding inquiry based learning into programming via paired assessment," *Innovation in Teaching and Learning in Information and Computer Sciences*, vol. 10, no. 1, pp. 72-77, 2011.
[16] C. Chang, C. Chung, and J. Chang, "Influence of problem-based learning games on effective computer programming learning in higher education," *Educational Technology Research and Development*, vol. 68, no. 5, pp. 2615-2634, 2020.
[17] J. Peng, M. Wang, and D. Sampson, "Scaffolding project-based learning of computer programming in an online learning environment," in *IEEE International Conference on Advanced Learning Technologies, ICALT 2017, Timisoara, Romania, July 3-7, 2017,* pp. 315-319.
[18] J. Bishop and M. Verleger, "The flipped classroom: a survey of the research," in *ASEE Annual Conference & Exposition, Atlanta, Georgia, June 23-26, 2013,* pp. 23.1200.1-23.1200.18.
[19] D. Berrett, "How 'flipping' the classroom can improve the traditional lecture," *The Chronicle of Higher Education*, vol. 12, no. 19, pp. 1-3, Feb. 2012.
[20] B. Tucker, "The flipped classroom," *Education Next*, vol. 12, no. 1, pp. 82-83, 2012.
[21] M. Guzdial, "Programming environments for novices," in *Computer Science Education Research*, S. Fincher and M. Petre, Eds. Lisse, The Netherlands: Taylor & Francis, 2004, pp. 127–154.
[22] J. Boustedt, A. Eckerdal, R. McCartney, J. E. Moström, M. Ratcliffe, K. Sanders, and C. Zander, "Threshold concepts in computer science: do they exist and are they useful?," in *ACM SIGCSE Bulletin*, vol. 39, no. 1, 2007, pp. 504–508.
[23] S. Anwar and M. Menekse, "A systematic review of observation protocols used in postsecondary STEM classrooms," *Review of Education*, vol. 9, no. 1, pp. 81-120, 2021. Available: https://doi.org/10.1002/rev3.3235
[24] M. Menekse, X. Zheng, and S. Anwar, "Computer science students' perceived needs for support and their academic performance by gender and residency: An exploratory study," *Journal of Applied Research in Higher Education*, vol. 12, no. 5, pp. 1025-1044, 2020. Available: https://doi.org/10.1108/JARHE-07-2019-0194
[25] I. Villanueva and S. Anwar, "Situating the place of multi-modal approaches place in engineering education research," *(Guest Editorial) Journal of Engineering Education*, vol. 111, no. 2, pp. 277-282, 2022. Available: https://doi.org/10.1002/jee.20460