Evaluating the Effectiveness of Flipped Classrooms for Teaching CS1

Ashish Amresh, Adam R. Carberry and John Femiani
College of Technology & Innovation
Arizona State University
Mesa, AZ, USA
ashish.amresh@asu.edu; adam.carberry@asu.edu; john.femiani@asu.edu

***Abstract*—An alternative to the traditional classroom structure that has seen increased use in higher education is the flipped classroom. Flipping the classroom switches when assignments (e.g. homework) and knowledge transfer (e.g. lecture) occur. Flipped classrooms are getting popular in secondary and post-secondary teaching institutions as evidenced by the marked increase in the study, use, and application of the flipped pedagogy as it applies to learning and retention. The majority of the courses that have undergone this change use applied learning strategies and include a significant "learning-by-doing" component. The research in this area is skewed towards such courses and in general there are many considerations that educators ought to account for if they were to move to this form of teaching. Introductory courses in computer programming can appear to have all the elements needed to move to a flipped environment; however, initial observations from our research identify possible pitfalls with the assumption. In this work in progress the authors discuss early results and observations of implementing a flipped classroom to teach an introductory programming course (CS1) to engineering, engineering technology, and software engineering undergraduates.**

Keywords – flipped classroom, introductory programming, learning with video, computing self-efficacy

I. INTRODUCTION

The ability to write and understand computer programs has become an essential skill for engineers to learn. The increased reliance on computer programming to address engineering needs has rapidly increased the number of programming-based courses required to receive any engineering degree. There has been a trend towards trying out newer strategies to enhance and improve how first and second year engineering students understand the principles of programming. Among the many strategies, the *flipped classroom* or *inverted classroom* [3, 4] has gained use being applied heavily at various secondary and post-secondary institutions.

The simplest form of a flipped classroom involves replacing traditional in-class lectures with video tutorials that students are expected to watch prior to class. The students are then required to come prepared for class, which is spent solving hands-on problems that are related to the video. Some variants of the flipped classroom model have students completing assignments, projects and homework in-class based on the watched video lectures. Inspiration and rapid increase of using the flipped classroom approach and many of its variants stem from the emergence of massive open online courses (MOOCs) [5], which have now been established around the world. This movement was spearheaded by the MIT OpenCourseWare [6]; aimed specifically to provide learning and knowledge to anyone who had a desire and motivation to gain new knowledge. The emergence of MOOCs as a ready-to-use learning platform and the popularity of the flipped classroom model have had a combined effect that has led to a vast migration of traditionally taught courses at many educational institutions.

This work in progress is a first attempt at a thorough evaluation regarding the effectiveness of the flipped classroom approach, specifically how CS1 or introductory computer programming is taught. Several authors have experimented with flipped classrooms in order to teach computer programming, but past assessments have focused on upper division courses. Gehringer and Peddycord [1] share their experience with using the flipped model to teach a junior level computer architecture class. Mason, Shurman and Cook [2] compare the effectiveness of a flipped classroom in an upper division engineering course that emphasizes problem solving. Both studies note that a major detractor of the flipped classroom approach is the introduction of the model in later years can be a difficult prospect for the student who has already had two or more years of learning in the traditional way. That leads to the current study, which will investigate the following research questions:

1) *Can a flipped classroom be an effective model for teaching CS1 to first and second year students?*
2) *How does the flipped classroom impact student computing self-efficacy?*
3) *What associated value do students have toward a flipped classroom approach?*
4) *What variants of the flipped classroom approach effectively help students learn?*

This paper will present our preliminary findings with the intention of shedding light on how a flipped classroom model compares to a traditional lecture-style approach in an introductory programming course.

978-1-4673-5261-1/13/$31.00 ©2013 IEEE

II. Course Design

The course used in this preliminary study was taught to students enrolled in engineering, engineering technology and software engineering programs. The three sections of the course were designed to be the first foray into learning a programming environment. Two of the three sections underwent the flipped model (experimental) with the third section being taught using a traditional approach (control). The course was structured in such a manner that one experimental section used the flipped classroom model for the first half of the semester and then switched to a traditional approach after taking the midterm. The second experimental section started out traditionally and switched to the flipped approach after the midterm. The control section was traditional throughout. Students were randomly enrolled into any of these three sections.

The course content included an introduction to the Python programming language and covered sequence, selection, iteration, flow-control, branching, object creation and manipulation, as is common to most CS1 courses [8]. The hands-on activities for all the sections centered on creating game-based algorithms using Python and the PYGAME library [7]. PYGAME provided the instructors with the necessary hooks needed to make the course hands-on. This was intentionally chosen to transform the course with considerably more applied learning elements. All in-class assignments were game-based and the videos necessary for solving the in-class assignments were specifically recorded to achieve student success. A total of 22 videos were recorded; 11 were needed prior to the midterm. The final exam was given after all the videos had been watched and all assignments were completed. A total of ten assignments were given to each student; five before the midterm and five after the midterm. A student, depending on the section they were enrolled in, either did the first five or the last five assignments in-class. All students took the same midterm and final across all the three sections.

III. Courses Assessment

The common assessments across the three sections were a midterm and a final. Both were timed tests and contained a finite number of problems specific to the course. In addition to the common assessments, students were also given pre-post surveys to identify computing self-efficacy [10] and associated value toward the course.

A. Summative Assessment

The project evaluated the performance on assignments between the students who were in the flipped model classrooms with students who were in a traditional lecture-based classroom. Summative assessment included the performance on: 1) assignments before taking the midterm, 2) performance on assignments after taking the midterm, 3) the midterm, and 4) the final. Table 1 shows these represented as columns and the rows represent the instructional model that was being applied prior to taking any of the column items. The flipped row indicates the results from the experimental sections when they were flipped. The traditional row indicates the results for the experimental sections when they were traditional. The control row represents the class that used a traditional approach throughout the entire semester. Different pre and post-midterm assignments were used in this course explaining the absence of these scores in the table.

TABLE 1. Average Assignment, Midterm, and Final Scores

Model	Assignments Pre-midterm	Assignments Post-midterm	Midterm	Final
Flipped	81.4	65.5	71.4	81.7
Traditional	69.1	66.3	59.2	63.4
Control	-	-	69.4	62.3

Our summative assessments suggest that the flipped model produced higher average scores in the course. This was consistent across both experimental sections eliminating any possible group differences that may have resulted from the randomized student placement. The control section had lower scores in both the midterm and final in comparison to the flipped sections.

B. Computing Self-Efficacy & Value Toward a Flipped Classroom Approach

A pre and post computing self-efficacy survey was used to evaluate any impact the flipped classroom model had on student confidence [9]. The survey was administered to students in both the experimental and control sections. An overall analysis of students' computing self-efficacy from both classes revealed an increase from pre (M = 53.3) and post-scores (M = 71.8). A paired-samples t-test of the 39 students confirmed this difference to be significant [t (38) = -3.459, $p \leq 0.001$].

A formative evaluation of value was conducted to measure the students' associated value towards the course and also their attitude towards the flipped approach. Experimental classes were asked questions pertaining to the flipped classroom approach. Initial analysis revealed a lack of positive value toward the flipped model.

A correlation analysis between computing self-efficacy and students' perceptions of value toward the flipped classroom model was shown to be not significant. This suggests that the course improved computing self-efficacy, but that the flipped classroom pedagogical approach was not likely to have been the reasoning for this increase. There can be many factors that could be the reason for the flipped classroom approach playing a less important role in the mind of the students. The critical reasoning suggested by the students as part of the open-ended survey were:

1) Adapting to a flipped classroom approach from traditional lecture is overwhelming.
2) Viewing long static videos can be boring.
3) A time-constrained setting to complete assignments can be intimidating in the early stages of using this approach.

IV. Conclusions & Future Work

In this study we report initial findings from a mixed-methods based experimental course design aimed at evaluating the flipped classroom model for teaching CS1. Our results show that the flipped approach has promise in improving student scores, but that students found this new approach to be overwhelming and intimidating at times. We aim to formalize our experimental method over the next two iterations of the course in subsequent years. We will be performing a longitudinal study during this period and collecting data from these iterations. Our intention of this study is to completely transform our CS1 courses to flipped classrooms.

The preliminary results presented were able to show that the flipped classroom model positively impacted student scores. Student computing self-efficacy also improved, but since the flipped approach was not used for the entire duration of the class, there is a chance for improper correlation between the approach and self-efficacy. We plan to address by further testing in the classroom.

We have early insights regarding the various approaches to flipping a classroom. It is not completely clear at this juncture whether all parts of the course need to be flipped. A change is definitely required in the way the videos are recorded. The approach of video taping long lectures taught by various instructors could be negatively impacting students value toward the flipped classroom model. Additionally, an important unknown to investigate is how best to increase students engagement levels with the videos.

Acknowledgment

The authors would like to thank the Arizona State University College of Technology and Innovation's Scholarship Support and Enhancement program for funding this research.

References

[1] Gehringer, Edward F., and Barry W. Peddycord III. "The inverted-lecture model: a case study in computer architecture." Proceeding of the 44th ACM technical symposium on Computer science education. ACM, 2013.

[2] Mason, G.S., Shuman, T.R., Cook, K.E. "Comparing the Effectiveness of an Inverted Classroom to a Traditional Classroom in an Upper-Division Engineering Course," Education, IEEE Transactions on, 99,1.1,2013

[3] Gannod, Gerald C., Janet E. Burge, and Michael T. Helmick. "Using the inverted classroom to teach software engineering." Proceedings of the 30th international conference on Software engineering. ACM, 2008.

[4] Day, Jason A., and James D. Foley. "Evaluating a web lecture intervention in a human–computer interaction course." Education, IEEE Transactions on, 49.4,2006

[5] Buck, Tara. "The Massive Effect of MOOCs on Higher Education." Ed Tech Magazine, 24, 2013

[6] Abelson, Hal. "The creation of OpenCourseWare at MIT." Journal of Science Education and Technology, 17.2, 2008

[7] McGugan, Will. Beginning Game Development with Python and Pygame. Will McGugan, 2007.

[8] Koffman, Elliot B., Philip L. Miller, and Caroline E. Wardle. "Recommended curriculum for CS1, 1984." Communications of the ACM, 27.10, 1984

[9] Kolar, H., Carberry, A., & Amresh, A. "Assessing student computing self-efficacy: A pilot study." To be presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA, April 27-May 1, 2013

[10] Kolar-Quezada, H., Carberry, A., & Amresh, A. "Measuring computing self-efficacy." To be presented at the Annual American Society for Engineering Education Conference & Exposition, Atlanta, GA, June 23-26, 2013