

Manoj Joseph D Souza,
Paul Rodrigues
India

Investigating the Effectiveness of the Flipped Classroom in an Introductory Programming Course

DOI: 10.15804/ tner.2015.40.2.11

Abstract

A flipped classroom is a teaching-learning environment in which students self-learn content by watching video lectures outside of the classroom and engage in learner-centered activities in the classroom with the guidance of an instructor. Earlier research into programming education has shown that students in the traditional classroom find introductory programming courses hard. This experimental study compared the effectiveness of a flipped classroom (experimental group, N=48) with a traditional classroom (control group, N=52) in two areas: 1. programming self-efficacy; and 2. academic performance. The results show that the students in the flipped classroom increased programming self-efficacy and achieved higher grades.

Keywords: flipped classroom, programming self-efficacy, academic performance, introductory programming

Introduction

Traditional classroom, in which knowledge is transmitted to students through one-way discourse and manner (Foertsch, Moses, Strikwerda, & Litzkow, 2002), has been found ineffective, inefficient and irrelevant to contemporary students (Brunsell & Horejsi, 2013). Students find that information imparted during lectures may come too slowly or cover what they already know; weaker students have trouble in acquiring information rapidly or may lack prior knowledge to understand the presented topic (Goodwin & Miller, 2013). To address many of the issues of the traditional classroom, educators have recently introduced a new pedagogical approach called a 'flipped classroom', which offers an opportunity to

incorporate active learning methods in the classroom, while still covering necessary learning material. Research studies suggest that active learners show higher learning outcomes in terms of problem solving, time mastery and conceptual understanding (Chi, 2009).

The ability to write and understand programs is the most essential skill required from software engineers. Engineering education has paid increased attention to programming related courses so that students develop the conceptual understanding and master problem solving techniques. There has been a trend to introduce newer strategies to teach the principles of programming to students and one of them is a 'flipped classroom' (Zingaro, 2014). This study was undertaken with a primary motivation to explore the effectiveness of this pedagogy in the area of programming education.

Literature background

Flipped Classroom: A flipped classroom is a kind of blended learning environment, where students learn instructional content watching video lectures at home and what used to be homework is done in class, where teachers offer personalized guidance and interaction with other students, instead of lecturing. Lage et al. (2000) provide the simplest definition of the flipped or inverted classroom: "Inverting the classroom means that events that have traditionally taken place inside the classroom now take place outside the classroom and vice versa". Bishop and Verleger (2013) define the flipped classroom as one consisting of two parts: interactive group learning activities inside the classroom and direct computer-based individual instruction outside of the classroom.

There is a considerable amount of research in support of the flipped classroom. Flumerfelt and Green (2013) showed impressive learning achievement, behavioural improvement and increased interaction between teachers and students in a flipped classroom. Students in a flipped classroom become more conscious of learning, improve their learning and make connections to course content (Strayer, 2012; Mason, Shuman, & Cook, 2013).

Research shows three primary motivations for implementing the flipped classroom. First, the flipped classroom replaces lecture time in class for active, collaborative, and problem-based learning and thus helps in reinforcing the concepts learnt without sacrificing the content (Papadopoulos & Roman, 2006; Bland, 2006; Demetry, 2010). Second, the flipped classroom allows the instructor to design the course material in different ways to cater to the varying needs and learning styles

of students (Zappe, Leicht, Messner, Litzinger, & Lee, 2009; Lage, Platt, & Treglia, 2000). Third, the flipped classroom can motivate students to become self-learners and life-long learners (Bland, 2006).

Self-efficacy: According to Bandura (1986), self-efficacy is a person's confidence in their abilities to complete a certain task successfully. Self-efficacy beliefs are important influential factors of whether individuals will be able to put in effort on a task and continue performing the task coping with difficulties. The person with high self-efficacy beliefs attempts tasks and performs them even though the tasks might be difficult and challenging, while the person with low self-efficacy beliefs gives up difficult tasks easily. Self-efficacy is not a measure of skill but it signifies what persons believe they can do with the skills they possess (Bandura, 1997). Since self-efficacy beliefs affect individuals' approach to new challenges and contribute to performance, they influence the person's thought processes, motivation and behavior (Bandura, 1997).

Four factors that determine self-efficacy are: enactive mastery experience, vicarious experience, verbal persuasion, and psychological states, of which the most influential factor is the enactive mastery experience which indicates a person's experiences with the past failures and successes (Bandura, 1997; 1986). If the past successes enhance a person's self efficacy, the past failures lower it.

Research suggests that self-efficacy is one of the most important factors that affect students' academic performance (Arslan, 2013). Some research studies showed a direct positive correlation between self-efficacy and academic achievement (Klomegah, 2007; Richardson, Abraham, & Bond, 2012). Other research showed that self-efficacy was a significant predictor of academic achievement (Azar, 2013; Coutinho, 2008).

Research questions and hypotheses

The aim of this experimental study was to investigate the effect of the implementation of a flipped teaching methodology on the learning outcomes and programming self-efficacy of students of an introductory programming course. This study had one independent variable: the teaching method (traditional classroom or flipped classroom) and two dependent variables: 1. academic performance and 2. students' programming self-efficacy. The study addressed the following research questions.

1. Will the flipped teaching method improve students' programming self-efficacy in an introductory programming course?

2. Will the flipped teaching method improve students' programming self-efficacy compared to the traditional lecture-based teaching method in an introductory programming course?
3. Will the flipped teaching method improve students' academic performance compared to the traditional lecture-based teaching method in an introductory programming course?

Based on the above research questions, the investigators formulated the following research hypotheses.

H1: Flipped teaching will improve students' programming self-efficacy beliefs in an introductory programming course.

H2: Flipped teaching will improve students' programming self-efficacy beliefs compared to the traditional lecture-based teaching method in an introductory programming course.

H3: Flipped teaching will improve students' academic performance compared to the traditional lecture-based teaching method in an introductory programming course.

Methodology

The traditional lecture-based method was used in the first section of the first year students of Computer Science and Engineering. The class was composed of 52 students, of whom 22 were male and 30 female, who served as a control group for this experimental study. Flipped teaching was conducted in the second section of the first year students of Computer Science and Engineering. The flipped classroom was composed of 48 students, of whom 26 were male and 22 female, who served as an experimental group. Computer Programming, an introductory programming course, was taught in both the sections by the same instructor. It was a three-credit course, which lasted 15 weeks. The students in the traditional classroom had three classes per week, each class lasting fifty minutes and a programming laboratory session of three hours. In the flipped classroom, the students had three programming sessions per week, each lasting 110 minutes.

Flipped programming classroom

Each student was given the relevant video lectures on Computer Programming by the instructor to be watched outside of the class time. The instructor selected

appropriate videos available from YouTube, using the criteria of duration, clarity and content. The duration of each video was chosen to be between 5 to 15 minutes, as students do not have patience to watch long videos. To ensure that all the students would follow audio, the videos were selected in such a way that they had subtitles in English. Every video clip covered a topic of the course material included in the syllabus.

In the flipped model, all the classes were conducted in the programming laboratory. The students sat in pairs in front of each terminal. The instructor allowed the students to choose their own pairs for the sake of compatibility. However, once the pairs were formed, they had to remain in them throughout the course. The class time was divided for various in-class activities. At the beginning, the instructor revised the objectives of the class. The instructor then gave time for the students to ask questions, clarify doubts on the video lectures they had watched at home. The students spent a large amount of time (approximately 80 minutes) doing programming exercises assigned by the instructor. Unlike the traditional laboratory where the students did all the programming exercises individually, in the flipped model the students used the pair-programming strategy to do the programming exercises. The instructor conducted a short quiz, which normally contained ten multiple choice questions to ascertain whether the students had watched the lecture videos before coming to class and to find out the students' learning progress. Before concluding a class, the instructor gave learning objectives for the next class so as to excite interest in the students to watch lecture videos at home. The students were also given lecture notes and reference material and some homework exercises. Table 1 presents the model of the flipped programming classroom.

Table 1. Flipped programming classroom

In class	Outside of the class
<ul style="list-style-type: none"> • Instructor revises goals and objectives of the class (5 mins) • Students ask questions and clarifications (10 mins) • Students do programming exercises in pairs (80 mins) • Students answer a short quiz (10 mins) • Instructor specifies goals and objectives for the next class (5 mins) 	<ul style="list-style-type: none"> • Students watch videos • Read lecture notes / reference material • Solve homework problems

Assessment methods: Student learning was evaluated by both internal as external assessment. The students from both the control group and experimental group were given three internal assessments after the 4th, 8th and 12th week, which contributed to 20% and a final exam, which contributed to 80% of the total mark. The design and evaluation of the internal assessment was done by the instructor, whereas the question paper setting and evaluation of the final exam was done by an external examiner and evaluator respectively.

The management of the college had randomly assigned the students of Computer Science and Engineering to two sections. To assess the similarity between the experimental and control groups before the experimental study, the students' performance in class, 12 examination as well as the pretest scores of programming self-efficacy were compared. Statistical analysis showed that there were no significant statistical differences between the two groups, indicating that both the flipped and traditional groups were similar in academic performance and self-efficacy beliefs.

Instrumentation: Data concerning the self-efficacy of the students of Computer Programming was collected through C Programming Self-Efficacy Scale (CPSES), developed from the Computer Programming Self-Efficacy Scale by Ramalingam and Wiedenbeck (1998). Their instrument was validated by giving the test to 421 students during the first week of the semester. Factor analysis identified four factors which Ramanlingam and Wiedenbeck (1998) labelled: (1) independence and persistence, (2) complex programming tasks, (3) self-regulation, and (4) simple programming tasks. The alpha reliability of the scale was 0.98 and the corrected item-total correlations ranged from 0.5 to 0.84.

CPSES was administered to both the experimental and control groups in the first week (pre-test) of the semester and also in the 15th week (post-test) of the semester. The participants were to rate their confidence in doing some specified C programming related tasks. The scale contained 29 items and the students had to rate their confidence for each item in a seven-point Likert scale ranging from 1 to 7 i.e., 'not confident at all' to 'absolutely confident'. The alpha reliability of the scale was found to be 0.94. The item-total correlation the self-efficacy varied from 0.42 to 0.86.

Results

The first research question: To find out whether the flipped teaching will improve students' programming self-efficacy in an introductory programming

course, a paired-samples t-test was done to compare the students' programming self-efficacy mean scores before and after implementing the flipped strategy. Analysis was made across four factors: independence and persistence, complex programming tasks, self-regulation and simple programming tasks. Significant differences were observed in the mean scores of the students' pre-test scores and post-test scores across all the four factors. The overall analysis showed that the post-test mean score ($M=3.83$, $SD=0.45$) was higher than the pre-test mean score ($M=2.63$, $SD=0.52$) at the 0.05 level of significance $t(47)=14.8$, $p<0.001$. The results confirmed the hypothesis (H1) that the flipped teaching would improve students' programming self-efficacy in an introductory programming course. Table 2 summarizes these results.

Table 2. Paired-samples t-test results of the four factors of students' programming self-efficacy before and after the use of the flipped teaching model

Factor	Pre-test mean	Pre-test SD	Post-test mean	Post-test SD	Mean difference	t-value	p
Independence and persistence	2.52	0.90	3.85	0.71	1.33	9	0.000
Complex programming tasks	2.25	0.70	2.63	0.57	0.38	4.3	0.000
Self-regulation	2.63	0.67	3.96	0.71	1.33	10.5	0.000
Simple programming tasks	3.08	0.87	4.87	0.83	1.79	11.1	0.000
Overall	2.63	0.52	3.83	0.45	1.2	14.8	0.000

The second research question: To find out whether the flipped teaching model will improve students' programming self-efficacy compared to the students taught in the traditional lecture-based model in an introductory programming course, an independent samples t-test was done to compare the programming self-efficacy scores of these two models. The data was collected from both groups in the last week of the semester. The results showed that there were significant differences between the mean scores of the flipped classroom ($M=3.83$, $SD=0.45$, $N=48$) and the traditional classroom ($M=3.38$, $SD=0.49$, $N=52$) at the 0.05 level of significance $t(98)=4.7$, $p<0.001$. These results confirmed the second hypothesis (H2).

The third research question: To determine if the flipped teaching strategy improved the academic performance of the students as compared with the students of the traditional classroom, an independent samples t-test was done, comparing the exam marks of the students of the flipped classroom and the traditional class-

room. Statistical analysis showed that there were significant statistical differences between the mean marks of the flipped classroom ($M=65.9$, $SD=11.72$, $N=48$) and the traditional classroom ($M=60.4$, $SD=12.54$, $N=52$) at the 0.05 level of significance $t(98)=2.28$, $p<0.01$. These results confirmed the third hypothesis (H3) that the flipped classroom students performed academically better than those of the traditional classroom.

Discussion

The results clearly show that the students in the flipped classroom increased their programming self-efficacy. A closer look at different factors reveals some important information. The overall programming self-efficacy measured during the pre-test showed that the students were little confident (*mean 2.63 out of 7*) before the course began. A possible interpretation of this could be that some students had not studied computer programming in their higher secondary school, as observed by the instructor. It could be also that even those students who had learnt computer programming, had not developed algorithmic thinking and problem-solving strategies but merely memorized worked-out examples to pass the examination. The results show an increase in self-efficacy in all the four areas: independence and persistence, complex tasks, self-regulation and simple tasks. However, there was the least increase in programming self-efficacy in complex programming tasks (*mean 0.38*) and the highest increase in simple programming tasks (*mean 1.79*). From Bandura's theory we can explain that the students gained mastery experiences that gave them a sense of accomplishment when they solved simple tasks which in turn increased their self-efficacy significantly, but on the other hand, lack of such mastery experiences in complex tasks failed to do the same.

It may be observed that the students' programming self-efficacy in the flipped classroom was significantly higher than that of the students in the traditional classroom after the completion of the course. A probable explanation could be that the students in the flipped classroom spent more time doing practical exercises that promoted their cognitive engagement and enabled them to interact efficiently with the learning material than the students in the traditional classroom. This explanation is consistent with the results of earlier research on self-efficacy revealing that enactive mastery experiences gained through performing a task while applying knowledge and skills are the most powerful source of developing a strong sense of self-efficacy (van Dinther, Dochy, & Segers, 2011).

A significant finding of this research study is that the flipped model has a potential to improve students' academic performance in an introductory programming course. These results support other earlier findings with different population and different subject areas (Sparks, 2013). A possible explanation of this result is that the students in the flipped classroom had opportunities to work together collaboratively on authentic, hands-on activities. Prior research on the flipped classroom reports that increased learning outcomes are due to the additional opportunities students get in the flipped classroom (Strayer, 2012).

Some earlier research studies showed that the flipped classroom was more suitable for upper division engineering courses, as the flipped format may be hard and difficult for students who have not developed strong study skills (Mason, Shuman, & Cook, 2013). Mason et al. (2013) report that there is a lack of research pertaining to the applicability of the flipped classroom to a first-year course and suggest future studies should investigate into this research gap. This study reports success of implementing the flipped classroom in a first-year course.

Conclusion

The important findings of this study are that the flipped classroom has a great potential to improve students' programming self-efficacy in an introductory programming course as well as students' academic performance. The study showed that the flipped classroom strategy can be applied in the first-year university courses where students are new to university education and have no prior experience of this method. The study also showed that the flipped teaching method is suitable for introductory programming courses, where students find programming difficult. The experimental results confirm the relevance of self-efficacy to the gaining of programming skills and support Bandura's theory.

Engineering education requires graduates who demonstrate strong problem-solving skills, independence and persistence, self-regulation and team spirit. These are the skills industry looks for while recruiting them. The flipped programming classroom offers a great opportunity to students to develop these skills.

References

- Arslan, A. (2013). Investigation of relationship between sources of self-efficacy beliefs of secondary school students and some variables. *Educational Sciences: Theory and Practice*, 13 (4), 1983–1993.

- Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8 (1), 26–32.
- Azar, F. (2013). Self-efficacy, achievement motivation and academic procrastination as predictors of academic achievement in pre-college students. *Proceeding of the Global Summit on Education*.
- Bandura, A. (1989). Regulation of cognitive processes through perceived self-efficacy. *Developmental Psychology*, 25 (5), 729–735.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: Freeman.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice Hall.
- Bishop, J., & Verleger, M. (2013). The flipped classroom: A survey of the research. *ASEE National Conference Proceedings*. Atlanta, GA.
- Bland, L. (2006). Apply flip/inverted classroom model in electrical engineering to establish life-long learning. *ASEE Annual Conference*. Chicago, IL.
- Brunsell, E., & Horejsi, M. (2013). A flipped classroom in action. *The Science Teacher*, 80 (2).
- Coutinho, S. (2008). Self-efficacy, metacognition, and performance. *North American Journal Of Psychology*, 10 (1), 165–172.
- Demetry, C. (2010). Work in progress—An innovation merging ‘Classroom flip’ and team-based learning. *Proceedings of 40th ASEE/IEEE Frontiers Education Conference*, (pp. T1E-1). Arlington, VA.
- Flumerfelt, S., & Green, G. (2013). Using lean in the flipped classroom for at risk students. *Educational Technology and Society*, 16 (1), 356–366.
- Foertsch, J., Moses, G., Strikwerda, J., & Litzkow, M. (2002). Reversing the lecture/homework paradigm using eTEACH® web-based streaming video software. *Journal of Engineering Education*, 91 (3), 267–274.
- Goodwin, B., & Miller, K. (2013). Research says / evidence on flipped classrooms is still coming in. *Technology Rich Learning*, 70 (6), 78–80.
- Klomegah, R. (2007). Predictors of academic performance of university students: An application of the goal efficacy model. *College Student Journal*, 41 (2), 407–415.
- Lage, M., Platt, G., & Treglia, M. (2000). Inverting the classroom: A gateway to creating an inclusive learning environment. *The Journal of Economic Education*, 31 (1), 30–43.
- Mason, G., Shuman, T., & Cook, K. (2013). Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper division engineering course. *IEEE Transactions on Education*, 56 (4), 430–435.
- Papadopoulos, C., & Roman, S. (2006). Implementing and inverted classroom model in engineering statics: Initial results. *Proceedings of ASEE Annual Conference*, (pp. AC2010–1868). Louisville, IL.
- Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19 (4), 367–381.
- Richardson, M., Abraham, C., & Bond, R. (2012). Psychological correlates of university

- students' academic performance: A systematic review and meta-analysis. *Psychological Bulletin*, 138 (2), 353–387.
- Sparks, R. (2013). Flipping the classroom: An empirical study examining student learning. *Learning in Higher Education*, 9 (2), 65–70.
- Strayer, J. (2012). How learning in an inverted classroom influences cooperation, innovation and task orientation. *Learning Environments Research*, 15 (2), 171–193.
- Zappe, S., Leicht, R., Messner, J., Litzinger, T., & Lee, H. (2009). Flipping the classroom to explore active learning in a large undergraduate course. *Proceedings of ASEE Conference*, (pp. AC2009–92). Austin, TX.
- Zingaro, D. (2014). Peer instruction contributes to self-efficacy in CS1. *Proceedings of the 45th ACM technical symposium on Computer Science Education* (pp. 373–378). ACM.