

Switching to Blend-Ed: Effects of Replacing the Textbook with the Browser in an Introductory Computer Programming Course

Patrick Seeling

Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
pseeling@ieee.org

Abstract—We describe the evolution of an introductory programming course into an active learning format, thereby replacing the traditional textbook and online tutoring system with an integrative version. We find a strong positive correlation between the student achievement scores in the integrated online textbook exercises and quizzes as well as overall course achievement. However, we observe that the comparable performance is not across all learning assessment categories when comparing subsequent course offerings. Particularly, we note slightly lower average performance for programming assignments and quizzes in the flipped classroom offering. For this active learning format offering, we perceive that performing students have a higher average score in the interactive textbook environment and skipped fewer overall assignments than the lower performing learners. We additionally find that students performing well in the online self-guided hands-on programming exercises perform significantly better than their lower performing peers.

Index Terms—Computer science education; Integrated textbooks; Programming; Learner performance

I. INTRODUCTION

Computer science education approaches have received much interest in recent years, as a tremendous push towards educating and graduating more students fluent in information technologies is made. Special considerations oftentimes are made with respect to initial introduction to programming courses, oftentimes offered to the general student population as on-ramping experience for a potential future computer science major. Several challenges exist that are reflected in the manners by which the traditional approach to instruction has been modified in order to increase learner performance.

Prior research indicates that a more interactive instructional approach can yield significant performance increases [1]. The resulting notion of classroom “flipping” has enjoyed much popularity due to the benefits that stem from increased student interactions, albeit not without potential drawbacks [2]. Specific considerations for the computer science instructional environment can be found in, e.g., [3], [4]. In addition to these modified instructional delivery methods, feedback during skill acquisition in computer science has emerged as an effective, yet difficult to achieve concept [5]. As introductions to programming oftentimes include learning a new programming

language paired with computational thinking on top of new tool usage, students could easily be overwhelmed. Practicing on a large scale is hampered by the common lack of resources necessary to continuously evaluate student code, especially in larger classes. Interactive online tutoring systems that provide self-guided hands-on exercises with feedback are a potential remedy for this problem [6], [7].

In this paper, we describe the learner performance differences by switching from a regular class offering to an active learning format in an introductory programming class at a large university in the Upper Midwest. While the initial course offering already included a nowadays common approach to foster interaction in the lecture by employing classroom response systems, the revised iteration of the course completely removed those and was offered in a dedicated re-configured classroom. The active learning format course revision additionally abandoned the traditional course textbook, separate online training exercises, and separate hands-on laboratory programming units for a continuous integrated hands-on experience that is complemented with a revised online version of the course textbook. The revised online book features integrated self-check exercises next to automatically graded assignments with feedback, similar to, e.g., [8]. The underlying educational content, exercises, and other course delivery facets remain constant, enabling the comparability.

The remainder of this paper is structured as follows. In the following section, we review the course offerings and their differences. Subsequently, we analyze our findings in Section III for different factors of influence before we conclude in Section IV.

II. COURSE OVERVIEW

The course under consideration is an introductory course for students in the Computer Science and Information Technology major programs offered by the Department of Computer Science, but it is also open to students across campus as part of the general education requirements. The course is based on the JAVA programming language (imperatives-first) and employs a standard textbook [9], typically covering selective

content that ends at the complexity level of multi-dimensional arrays. In the Spring semester offering, the physical textbook was complemented by MyProgrammingLab (MPL), which is an online tutoring environment that allows students to submit short programs for evaluation and feedback. In the Fall course iteration, the physical textbook and separate MPL were merged into an integrated environment (REVEL), which combines the online textbook with inline hands-on exercises that match those of the former MPL environment.

While the Spring semester featured separate lectures and laboratory sessions with guided hands-on exercises as part of the learners' experience, the Fall course offering abandoned this lecture/lab combination in favor of an integrative active learning course delivery format. This course modification is typical for the "flipped" class approach. In the active learning format, the students experienced continuous hands-on exercises similar to the ones experienced in the Spring course offering, making both iterations comparable. In order to derive maximum utility from the active learning sessions, however, students had to work through the immersive textbook themselves. Furthermore, the active learning course offering utilized a specially designed instruction room, which features group tables and monitors in addition to an instructor station.

Each course iteration was offered by the same instructor and employed the same grading schema. Individual items were graded on a pass/fail basis that was augmented by tokens for participation in special exercises within the Blackboard learning management system employed by the university. For example, tokens could be employed for skipping an assignment or getting extra time for completion. The course's main grading categories included one mid-term and one final course examination as well as multiple online quizzes, programming homework assignments, and assigned exercises in the MPL/REVEL environments. For all programming assignments, including those being part of the examinations, a dedicated and communicated rubric was employed. Quizzes and other examination components are based on multiple choice/select question types that covered theoretical questions (i.e., those typically stemming from the textbook) and program code understanding questions. While quizzes were administered using the learning management system and allowed for one to two repetitions, the assigned MPL/REVEL exercises allowed for unlimited repetition and were only scored based on final correct answers to allow students to correct their potentially mismatched conceptual understanding.

III. RESULTS

We now discuss the results obtained by different viewpoints for both outlined course offerings, which were made comparable and re-scaled to relative achievement scores.

A. Overall Differences Between Offerings

We initially evaluate the differences for the learners' achievements for their overall attained semester performance, as well as for individual categories (programming assignments, self-guided hands-on training in MPL/REVEL, exams, and

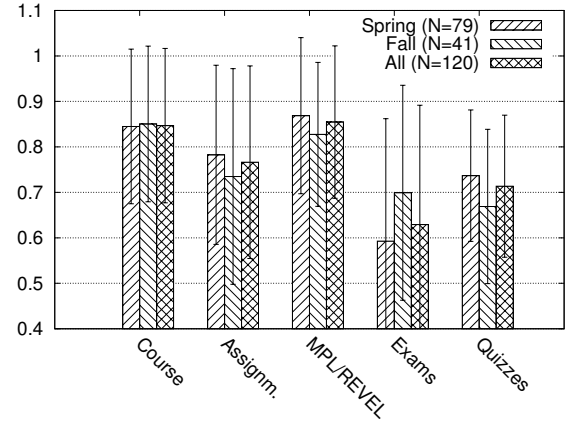


Fig. 1. Relative performance for the entire course, as well as programming assignments, self-guided hands-on training, exams, and quizzes for N=79 students in Spring and N=41 students in Fall.

quizzes). We illustrate these relative performances in Figure 1 and compare differences between the two consecutive course offerings. A beginning evaluation of the variance in each category employing Levene's statistic reveals that one-way ANOVA can directly be applied. Initially, we do not find a significant difference for the total course performance between the two semesters [$F(1, 118)=.028, p=.868$] or the relative programming assignment performance [$F(1, 118)=1.373, p=.244$]. We similarly do not observe a significant difference in the learners' embrace of the MPL or REVEL online training parts, $F(1, 118)= 1.639, p=.203$. We find a significant difference for the learners' relative examination scores as determined by one-way ANOVA, $F(1, 118)= 4.568, p=.035$. Students in the MPL offering ($M=59.28\%, SD=26.92\%$) were outperformed by students in the "flipped" REVEL course offering ($M=69.92\%, SD=23.63\%$). While we do not find a significant difference for the mid-term examination, $F(1, 118)= 1.167, p=.282$, the comprehensive final semester examination ($M=1.797, SD=.979$ in Spring vs. $M=2.244, SD=.83$ in Fall) exhibits a significant difference, $F(1, 118)= 6.204, p=.014$. For the online quizzes, ANOVA also reveals a significant difference between the two consecutive course offerings, $F(1, 118)= 5.277, p=.023$. Here, however, the students within the regular class ($M = 73.68\%, SD=14.45\%$) outperformed those in the active learning one ($M=66.89\%, SD=16.97\%$).

In summary, we note that the major differences that stem from the course modifications to the "flipped" format are the increased comprehensive exam performance paired with a decreased online quiz performance while overall course achievements and those attained for the programming assignments and hands-on programming training remain comparable.

B. Differences by Self-Guided Hands-On Performance

We now group the students in both course offerings into a low and a high performing group, based on the relative number of exercises skipped (or not correctly solved) in the online interactive tutoring modules (MPL/REVEL). The

TABLE I
RELATIVE LEARNER PERFORMANCE BY ACHIEVEMENT IN THE ONLINE
INTERACTIVE TUTORING ENVIRONMENTS (MPL/REVEL), ALL IN
PERCENT.

Component	High (N=60) <i>M, SD</i>	Low (N=60) <i>M, SD</i>	All (N=120) <i>M, SD</i>
Course	93.72, 10.31	75.64, 17.56	84.68, 16.97
Assignments	84.24, 14.20	69.02, 24.19	76.63, 21.18
MPL/REVEL	96.55, 0.24	74.37, 17.64	85.46, 16.77
Exams	67.22, 27.27	58.61, 24.64	62.92, 26.24
Quizzes	78.40, 12.71	64.32, 15.17	71.36, 15.62

median percentage that students were skipping this online training is 9.375 %. We present the results for the high, low and all students in Table I. As the variance for the overall course score is not homogeneous amongst the two groups [Levene's $F(1, 118)=9.144$, $p=.003$], we report the Welch F-ratio. There was a significant effect of the on-line programming tutoring system with overall course performance, $F(1, 95.333)=47.303$, $p < .001$. Similarly, the assignment average violated the homogeneity of variance assumption [Levene's $F(1, 118)=15.056$, $p < 0.001$], and employing Welch's F-ratio, we note a subsequent significant interaction, $F(1, 95.333)=17.66$, $p < .001$. For the average exam performance, we only approach significant differences, $F(1, 118)=3.294$, $p=.072$, with no higher significance to be found for the individual semester exams themselves. Lastly, we observe a significant effect for the average quiz performance revealed by a one-way ANOVA, $F(1, 118)=30.397$, $p < .001$.

In summary, we observe that learners who perform well in the self-guided hands-on exercises within the MPL or REVEL environments significantly outperform those that skip or fail at a significant fraction of these exercises. Most importantly, we observe this trend across all course grading categories.

C. Differences By Course Iteration and MPL/REVEL Performance

We now evaluate the individual groups in each semester offering, i.e., we determine how the relative performance of MPL/REVEL assignments effects the overall student performance in the remaining categories. In other words, this evaluation aims at the elimination of the effects that stem from the "flipped" classroom approach, illustrated in Figure 2. We initially observe that learners who perform well in the self-guided hands-on programming environments on average outperform in both semesters. Again, we report the Welch F-ratio as inhomogeneous variances were encountered for the overall course performance [Levene's $F(3, 116)=3.287$, $p=.023$]. There was a significant effect of the semesters' online programming tutoring system usage with overall course performance, $F(1, 48.442)=15.962$, $p < .001$. However, the significance can be observed within each semester between students that exhibit a high use of the separate MPL [$t(49.21)=5.127$, $p < .001$] or integrated REVEL [$t(38.915)=4.788$, $p < .001$] environments.

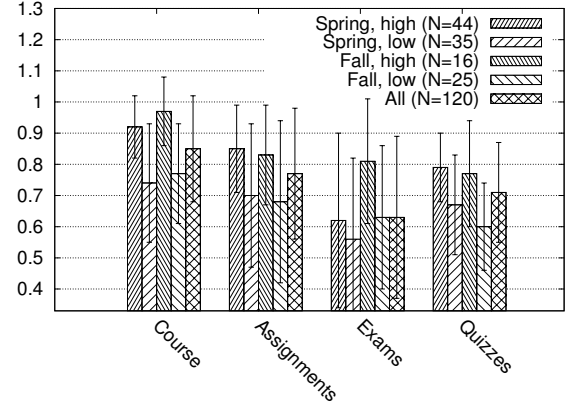


Fig. 2. Relative performance for the entire course, as well as programming assignments, self-guided hands-on training, exams, and quizzes for N=79 students in Spring and N=41 students in Fall.

A similar observation concerning the variance is obtained for evaluation of the programming assignments [Levene's $F(3, 116)=5.442$, $p=.002$], resulting in another significant effect based on the Welch F-ratio [$F(1, 47.143)=5.838$, $p=.002$]. Again, closer inspection of groups indicates that significant effects can be observed within each semester's MPL and REVEL users with $t(52.698)=3.383$, $p=.001$ and $t(38.914)=2.266$, $p=.029$, respectively.

For the overall exam performance, a one-way ANOVA results in a significant interaction between groups, $F(3, 116)=3.739$, $p=.013$. Specifically, we observe significant effects between the learners utilizing MPL or REVEL the most [$t(116)=-2.582$, $p=.011$] as well as within the high and low REVEL utilizers in the second semester [$t(116)=2.287$, $p=.024$].

Lastly, a significant interaction between groups is found for the relative quiz performance, $F(3, 116)=16.345$, $p < .001$. As in some of the prior evaluations, we observe no significant effects between the semesters, but within each semester. We note a significant difference between those students that perform well or less in the MPL [$t(116)=3.763$, $p < .001$] and REVEL [$t(116)=3.708$, $p < .001$] environments.

In summary, we observe that the finer-grained grouping by course offering and hands-on programming environment performance indicates that learners with high achievement in the hands-on environments significantly outperform their lower achieving counterparts in each semester, with little differences in between the groups by semester.

D. Correlations

Motivated by these strong indicators within each semester concerning the utilization of the online tutoring environments for computer science learners, we evaluate the Pearson correlations between the quizzes, programming assignments, and final semester scores in relationship to the learner achievement in the MPL/REVEL environments, illustrated in Figure 3.

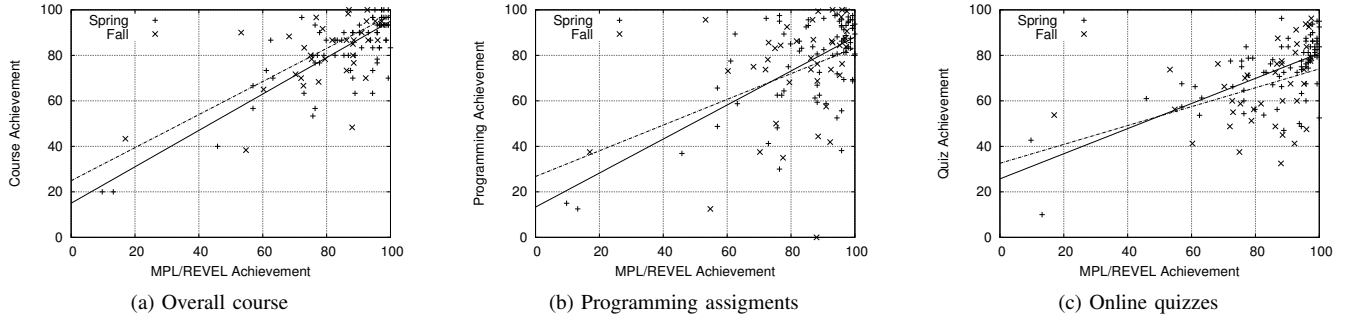


Fig. 3. Comparison of correlations between the automatically graded exercises in MPL/REVEL and attained student achievements for online quizzes, programming assignments, and total semester course performance.

We initially observe strong correlations between the overall course achievement and the performance in the MPL environment [$r(79)=.81, p < .001$] as well as the REVEL environment [$r(41)=.67, p < .001$]. While for the Spring semester, we note strong correlations for assignment and quiz performance with $r(79)=.65, p < .001$ and $r(79)=.66, p < .001$, respectively, we observe weaker correlations for the Fall semester with $r(41)=.38, p < .05$ and $r(41)=.39, p < .05$, respectively.

Interestingly, the strong semester performance correlation allows to abstract the impacts that the self-guided hands-on online programming tutorial component has. Assuming a direct correlation for the “regular” course, where a textbook is paired with the MPL environment, a “base” performance of about 15 % can be noted with each percent achievement in the self-guided hands-on environment accounting for almost 0.8 percent of the final semester score on average. Similarly, in the “flipped” course, where the textbook is online and self-guided hands-on exercises are included in the text, students start at a higher level of almost 25 % and each percent achievement in the REVEL environment could be attributed to 0.73 percent of the final semester score, on average. The active learning course iteration’s exhibition of a less pronounced correlation could be attributed to the impacts that the “flipping” has on the overall performance.

E. Performance Predictors

In absence of the perfect correlation we assumed in the previous discussion, we additionally evaluate a linear model to predict the semester performance based on the individual course grade components, again based on relative achievement. For each semester, we employ the automatic linear modeling in SPSS to determine the coefficients as well as their importance (after outlier elimination), with results provided in Table II. These individual values are not directly aligned with the overall weights that stem from course achievement through points in each category (approx. 26 %, 20 % for quizzes), whereby the programming assignments consistently outweigh the other items. Due to the previously discussed interactions, we note the significant interplays reflected here as well. Interestingly, we note a shift in the importance of the self-guided hands-on exercises, which increase from 17.4 % to

TABLE II
LINEAR MODEL PARAMETERS FOR SPRING AND FALL OVERALL COURSE PERFORMANCE BASED ON BASIC COMPONENTS. INDIVIDUAL SIGNIFICANCE WAS $p < .001$ EXCEPT FOR THE FALL INTERCEPT, WHICH WAS $p = .409$

Component	Spring (MPL)		Fall (REVEL)	
	Coeff.	Import.	Coeff.	Import.
Intercept	-17.062		-4.445	
Assignments	0.399	0.396	0.327	0.368
MPL/REVEL	0.382	0.174	0.374	0.230
Exams	0.187	0.275	0.171	0.135
Quizzes	0.343	0.155	0.328	0.266

23 % and quizzes increasing by 10 % with a similar drop in exams. The “flipping” of the course has, in turn, resulted in an additional importance of the hands-on practices and quizzes, while the better exam performance reduces that item’s importance.

This is readily corroborated if the nature of the active learning environment is taken into account: Theoretical foundations are only recaptured in hands-on activities in class, whereas the actual studying and initial conceptualization by learners takes place outside the classroom (and is fostered by the self-guided hands-on exercises embedded into the course book and reinforced through online quizzes).

IV. CONCLUSION

We find a significant effect for learners utilizing the interactive tutoring systems employed in both course iterations. We observe that students performing well in hands-on online programming exercises in either mode employed (separate textbook and MPL vs. immersive REVEL) surpass their lower performing counterparts in all grading categories. While in line with general prior research, the notable finding of our study includes the importance shift in grading categories from exams in the traditional delivery format to the online hands-on tutoring system as well as online quizzes as course performance predictors.

Continuing this current research, we will investigate the effects that stem only from the switch to the integrative textbook within future course offerings.

ACKNOWLEDGMENTS

We thank the team at Pearson for their help with the configuration and evaluation of the MPL/REVEL environments and C. Cooney for her help in the data analysis.

REFERENCES

- [1] B. Simon, J. Parris, and J. Spacco, “How we teach impacts student learning: peer instruction vs. lecture in cs0,” in *Prof. of ACM SIGCSE*. Denver, CO, USA: ACM, Mar. 2013, pp. 41–46.
- [2] J. L. Bishop and M. A. Verleger, “The flipped classroom: A survey of the research,” in *Proc. of ASEE Annual Conference & Exposition*, Atlanta, GA, USA, Jun. 2013, pp. 1–18.
- [3] M. Lou Maher, C. Latulipe, H. Lipford, and A. Rorrer, “Flipped classroom strategies for cs education,” in *Proc. of ACM Technical Symposium on Computer Science Education (SIGCSE)*, Kansas City, MO, USA, Feb. 2015, pp. 218–223.
- [4] M. N. Giannakos, J. Krogstie, and N. Chrisochoides, “Reviewing the flipped classroom research: reflections for computer science education,” in *Proc. of the Computer Science Education Research Conference (CSERC)*. Berlin, Germany: ACM, Nov. 2014, pp. 23–29.
- [5] C. Ott, A. Robins, and K. Shephard, “Translating Principles of Effective Feedback for Students into the CS1 Context,” *ACM Transactions on Computing Education (TOCE)*, vol. 16, no. 1, pp. 1–27, Jan. 2016.
- [6] K. Buffardi and S. H. Edwards, “Responses to adaptive feedback for software testing,” in *Proc. of the Conference on Innovation & Technology in Computer Science Education (ITiCSE)*, Virginia Tech. Uppsala, Sweden: ACM, Jun. 2014, pp. 175–170.
- [7] A. N. Kumar and L. C. Kaczmarczyk, “Programming tutors, practiced concepts, and demographics,” in *Proc. of the ASEE/IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, OK, USA, Nov. 2013, pp. 773–778.
- [8] R. H. Landau, M. J. Paez, and C. C. Bordeianu, “A blended, multimodal access eTextBook in computational physics,” in *Proc. of the Conference of the Extreme Science and Engineering Discovery Environment (XSEDE)*, Chicago, IL, USA, Jul. 2012, pp. 1–7.
- [9] Y. D. Liang, *Intro to Java Programming*. Pearson Education Limited, 2014.