Flipping the CS1 and CS2 Classrooms in Central Asia

Benjamin Tyler and Madina Abdrakhmanova
Department of Computer Science
Nazarbayev University
Astana, Kazakhstan
{btyler, madina.abdrakhmanova}@nu.edu.kz

Abstract— **In this paper, we report on our approach and preliminary observations of transitioning to a flipped classroom model of instruction for our first-year undergraduate programming courses. We believe that our situation is rather unique, in that we are doing so in the context of a newly-formed, Western-style university in Central Asia. Before the transition, our introductory programming course (in C), and our data structures course (in Java), were taught in a more traditional lecture-lab setting. Now, all lesson content is contained in online video lectures, which are embedded in our WordPress course pages. In-class time is now spent almost exclusively on students working through programming exercises in pairs, getting help from instructors and teaching assistants as needed.**

Our primary motivation in moving to a flipped classroom was to allow us to effectively cover material in more depth in our lectures, without sacrificing class time where we wanted students to focus more on hands-on activities. A secondary motivation was to allow us to deliver the course content in a consistent manner across multiple sections, from semester to semester. This has been a challenge given the steadily growing numbers of students in these courses, coupled with the continual additions and changes in faculty members and TAs.

As part of our preliminary study, we compared student performance in terms of final exam scores before and after the transition, and collected feedback from students using questionnaires. We did note a slight, though not significant improvement in exam scores. The student feedback was very positive, and indicated a high level of student involvement with the new approach.

Keywords—Flipped classroom; blended learning; Central Asia; Kazakhstan; Computer Science; Programming

I. INTRODUCTION

For the past several years, there has been an increased interest in employing alternative modes of instruction that leverage technology for university courses to improve student learning. In this paper, we describe our transition from a more traditional lecture/lab approach in teaching introductory programming, to one that uses the flipped classroom approach [1, 2, 3]. The flipped classroom is where students are provided with online course materials that they are expected to study outside of class, and in-class time is used for doing exercises related to the material. Students are able to go through the materials at a reasonable pace on their own time, while being able to get direct help from the instructor and TAs during the class exercises. In this way, the role of the classroom instructor shifts from one of lecturer, to one of facilitator and guide.

We believe that our situation in implementing the flipped classroom is interesting and unique: we are doing this at a relatively new, highly competitive, Western-style university in Central Asia. At Nazarbayev University, English is the language of instruction, although it is typically the third language of the students (Kazakh and Russian being the first two). Furthermore, the secondary education experience of the students generally reflects a post-Soviet style of instruction, as opposed to a more Western one.

In this paper, we present our approach to using the flipped classroom in our first-year programming courses, and provide the findings from our preliminary study. In the following section, we discuss the background and motivation behind our adopting the flipped classroom approach. In Section III, we briefly discuss related work, and then describe our approach in Section IV. Results of the preliminary study are presented in Section V, and future work is briefly discussed in Section VI.

II. BACKGROUND

A. About the University

Nazarbayev University (NU) was launched in 2010 by special decree of the government of Kazakhstan, and has been granted the unique status of an autonomous educational organization. This allows the university to develop policies, curriculum, and procedures outside those established by the government, facilitating the adoption of best international practices in higher education. NU has partnered with international institutions such as University College London, University of Wisconsin-Madison, University of Pittsburgh, Duke, Penn, and Cambridge. Admission is competitive, drawing from the best students in the country. NU currently has Observer status in the International Association of Universities, and will be eligible for full membership in 2017, once we graduate our third cohort of students.

B. The Courses

In Autumn semester of 2012, the CS department introduced two new first-year programming courses, CSCI 151 – Programming for Scientists and Engineers, and CSCI 152 – Performance and Data Structures. CSCI 151 is an introductory programming course using the C language, which focuses on control structures, loops, arrays, functions, pointers and memory allocation, along with searching and sorting. CSCI 152, which was initially taught in C++, but now in Java, focuses on object-orientation, abstract data types and their

implementation, binary search trees and hash tables, as well as basic algorithmic analysis and big-O notation.

Both of these courses are required of all undergraduate CS majors and minors, and initially CSCI 151 was only required of Math and Robotics majors, though strongly recommended for Physics students. Over time, however, these and other departments wanted their students to improve their programming skills, and so they either required or strongly encouraged their students to take these classes. Starting in Autumn 2016, both CSCI 151 and 152 will be required of all Math, Robotics, Physics, and CS majors. CSCI 151 will also be required for Chemistry and Biology majors as well.

C. Motivations for Using the Flipped Classroom

Our primary motivation for switching to the flipped classroom model was to improve student learning, and more effectively meet learning objectives. A guiding philosophy of the department has always been "learning by doing", and most of the courses in our curriculum emphasize hands-on work in and out of the classroom. In our introductory programming courses, we found that when using the traditional approach, we had to spend more than half of the class time lecturing and going through examples, leaving less time for in-class exercises. Using the flipped classroom approach frees up most of this time for such work, and gives instructors more opportunities to work with students one-on-one.

We had also found that we would have to slow the pace of our lectures to accommodate some students who struggled with English. By using online video materials, students would be able to replay clips that they may not understand on first viewing, and could go through the material with no time pressure.

Our secondary motivation was to help ensure effective and uniform delivery of first-year programming courses, given our limited faculty and TA resources. Though the use of the same online resources across different sections, and the ability to refine these common materials over time, we are able to work towards a consistent and gradually improving student experience.

Faculty recruiting has been a challenge, given the relative newness and lack of familiarity about the university outside of the region. As a result, the number of full-time CS faculty and teaching TAs has not grown significantly in the past four years. This is a significant problem given the growing number of students taking CS courses, and our introductory programming courses in particular (see Table I). Furthermore, many of the newer faculty and TAs who would be expected to teach our first-year courses are relatively inexperienced in the classroom, and may not be entirely familiar with the direction of these courses. Using pre-prepared materials enables these instructors to "ease in" to these courses.

TABLE I CS ENROLLMENTS V. NUMBEFR OF CS INSTRUCTORS

	2012-13	2013-14	2014-15	2015-16	2016-17
Total enrollment in CSCI 151 & 152	183	217	218	338	490 (projected)
Total enrollment in CS courses	467	770	926	1156	-
Avg. # of full-time CS faculty and teaching TAs	8	10	9	10.5	-

III. RELATED WORK

The flipped classroom model of instruction has been used to teach a variety of courses in different fields, including mathematics [4], engineering [5], business [6], and medicine [7]. Bishop and Verleger [8] provide a comprehensive overview of this pedagogical model in their survey, and focus on the educational theories behind it. Giannakos and Krogstie [9] present their own survey on the evaluation of the flipped classroom approach, and conclude that it improves students' engagement and overall performance, while noting the high initial cost of implementation and student adaptation.

In the area of Computer Science, Maher et al [10] report on flipping four different CS courses, and compares several strategies and approaches. They do note the high level of effort needed to create your own resources for a flipped course, and report on increased student engagement. Gehringer and Peddycord [11] apply the flipped approach for a computer architecture course, using previously recorded classroom lectures. While they noted higher levels of student engagement, they also saw that student performance on exams were lower in comparison to the non-flipped course. Jonsson [12] reports on a flipped introductory Object Oriented programming course, where recorded screencasts were used, which is similar to our approach. He saw that students in the flipped sections did much better on exams, although the participation was lower than expected.

In a non-western context, Lee and Lee [13] sought to determine whether the positive results of the flipped classroom experiments in the western world can be replicated in Asia. Similar to our approach, in their Data Structures course, students have to complete worksheets given at the beginning of class by the end of the period. Their study is ongoing, and they have yet to evaluate student performance, though they have found an increase in student participation. In Japan, Hayashi et al [14] used the flipped classroom model in two of their courses: Programming Skill (in C) and Object Oriented Programming (Java), which are very similar to the two courses we have transitioned. In comparison to the previous semesters using a more traditional approach, the students' performance improved and the students expressed more confidence in their programming skills.

IV. INITIAL APPROACH AND TRANSITION

A. Before the Transition

We began offering CSCI 151 and 152 during the 2012-13 academic year, and we refined the course over the next two years, where we used a common lecture/demonstration approach in the classroom. Our two classrooms are essentially computer labs, each having four large rectangular tables, with eight computers at each. Each classroom also has three projectors connected to a common DVI input. For these two courses, we used a common syllabus, and the instructors of the individual sections were encouraged to use the same basic materials. During this time, the courses were structured as follows:

Daily lessons with in-class exercises. At the beginning of most class periods, instructors would lecture and give live coding demonstrations using pre-prepared PowerPoint slides as

a guide during class time. This would often take around 60% of the class time on non-quiz days; the other 40% would be spent on students doing in-class exercises to be turned in at the end of the day. These in-class exercises would generally be graded on a course-grained (e.g., 0 to 3) scale.

In-class quizzes. Roughly 10 to 12 times during the 15-week semester, we would give brief 10-15 minute quizzes at the beginning of class. Initially, these were paper-based, but most instructors transitioned to using online, Moodle-based quizzes.

Sample programming assignments. Over the course of the semester, students would be assigned 5 or 6 graded programming assignments. Some of these assignments could be done in pairs. After submission, students would be required to present, answer questions about, and defend their work to an instructor or TA during office hours for "live grading". This was done to encourage student reflection on their work, and to act as a deterrent to plagiarizing code.

Midterm and Final Examination. A written, one-hour midterm exam at the middle of the semester, and a two-hour final examination were also used for student assessment. Students were not allowed to use notes or computers during exams. Together, these exams would account for about 50% of the student's final grade.

B. First Flipped Iteration – Moodle-Based Hosting

Initial planning for converting our first-year courses began during summer 2014, when we evaluated several different technologies that we could potentially use in creating and hosting course materials. It was decided that we would use Camtasia (https://www.techsmith.com/camtasia.html) for screen recording and editing, and any possible video capture. We decided in the first stage to host all of our course materials on Moodle, the learning management system used at Nazarbayev University.

During Spring semester 2015, we flipped all sections of both CSCI 151 and 152, creating video lectures from screen recordings that were taken *outside of class*. We did not use video capture of in-class lectures for this – instead, we recorded on-screen program coding demonstrations that were narrated, along with slide-based presentations and animations for exposition and summary. After about four weeks of refining our process, it would take approximately 4 hours to create the materials for a single day's lesson, which included screen recording and editing, creating animations, developing demonstration code examples, as well as creating the in-class assignment for that day. For the day's lesson, a single video file of about 20 to 30 minutes in length would contain about 6 separate clips, each covering one of the day's subtopics.

Although this approach is more time consuming than if we had just recorded our standard in-class lectures, we believe the benefits were worth it. In this way, we have greater flexibility in terms of presentation and editing of what the students see and hear. We also can cover the material in the desired depth without class time constraints. In order to keep students' from getting lost during an in-class lecture, either due to the complexity of the material or language issues, we often have to simplify the material, slow down, and/or repeat ourselves. This is not so much an issue with out-of-class recording, since students can watch, rewind, and re-watch individual clips when needed.

Class time was now structured such that the instructor would give a 5 to 10 minute summary of the key points from the lesson at the beginning, and then students would work alone or in pairs on a few programming exercises that would have to be submitted at the end of the class period. As before, we would also have occasional short quizzes at the beginning of class.

Student performance and feedback were very positive during the semester, so we decided to continue using the flipped classroom, and further refine our approach.

C. Second Iteration – Dedicated WordPress Webpages

During Spring 2015, we found that using Moodle to host all of our course materials was very inconvenient. We had to post the same materials for each individual section, and furthermore, we would have to do the same for each subsequent semester. It would be easier to set up and maintain one set of online materials that could be accessed by all students in a single place, and simply update them between semesters. At this time, we also discussed ways to improve the organization and navigability of the materials, and make the interface more visually appealing. We decided on using WordPress (https://wordpress.com/), a free, open-source content management system, for creating dedicated webpages for both of the courses. Due to infrastructure and firewall issues, we chose to host the new course webpages on a third-party server.

During Autumn 2015, we migrated the online materials for CSCI 151 to the new WordPress pages, and did the same for CSCI 152 during Spring 2016. The organization of the pages for an individual course includes separate pages for the course introduction, syllabus and schedule, a listing of the individual lessons, tool installation instructions, further resources, and a discussion forum.

Each video lesson from the previous iteration was split into their subtopic clips, refined for improved pacing and clarity, and then uploaded to YouTube as unlisted videos (videos that can be accessed by anybody, but do not show up in searches). Each daily lesson has its own individual page, where the main section contains embedded links to the video clips, along with lesson text between the videos to tie them together to form a cohesive narrative.

After the main lesson content, students can follow a link to an online practice quiz (created using the Wp-Pro-Quiz plugin for WordPress) to see how well they understand the key points from the lesson. At the bottom of the lesson page, several open-ended questions are listed which could be used to facilitate classroom discussion, and then a list of several potential in-class exercises.

Anyone can register and access the course webpages from the link http://sst-csci.com/. It should be noted that several non-students have been accessing these materials to learn how to program, or brush up on their skills.

The overall class time was structured in much the same way as with the previous iteration, except we eliminated the

quizzes for student evaluation, but did phase-in a second evening midterm examination. It should be noted that students would often begin working on in-class exercises *before* class, which we encouraged. By us providing the exercises early, students would have additional incentive to go through and review the materials, if needed, to help ensure that they could complete the day's exercise on time. If they struggled in making progress on the exercise before class, then they would be in a better position to ask questions during the actual class period.

V. Results

For our preliminary study, we wanted to get a rough assessment of how transitioning to the flipped classroom has impacted student performance and attitudes towards the new approach. At the same time, we also took into consideration other potential benefits such as the ability to instruct a larger number of students with limited faculty and TA resources.

For investigating student performance, we compared the average final exam scores for each semester, starting with Autumn 2013, which is similar to the approaches used in [11], [12], and [14]. We as a department have made an effort to keep the format, topical coverage, and grading rubrics relatively consistent for the final exams for our introductory programming courses. Changes in grading policies and evaluation criteria for other course elements have made them less useful for performing relative comparisons.

The results shown in the below tables do indicate slight, though not statistically significant improvements in average exam scores from the pre-flipped to the flipped semesters. If final exam scores are reliable indicators of student learning, then maintaining these same levels should still be seen as a positive outcome for the following reasons:

- The flipped classroom approach has allowed us to service a larger number of students by increasing class sizes, and enabling us to utilize faculty who are inexperienced in teaching first-year programming.
- The preparation time for classroom instructors has been considerably reduced.
- As of Autumn 2015, we are servicing a larger percentage of students from departments outside of Computer Science and Robotics, which traditionally do not perform as well in these courses. The below results show that this has not had a negative impact on average exam scores.

TABLE II CSCI 151 Average Final Exam Results

	Pre-Flipped			Flipped (Moodle)	Flipped (WordPress)	
Semester	Au '13	Sp '14	Au '14	Sp '15	Au '15	Sp '16
Avg. Exam Score	76.8	78.6	80.9	75.8	81.2	79.5
Exam Count	70	73	82	47	147	63

TABLE III CSCI 152 Average Final Exam Results

	Pre-Flipped		Flipped (Moodle)		Flipped (WordPress)
Semester	Sp '14	Su '14	Sp '15	Su '15	Sp '16
Avg. Exam Score	81.5	74.2	79.5	80.6	81.6
Exam Count	44	20	55	25	96

TABLE IV Selected Questionnarie Results

	Spring 2015		Spring 2016	
	CSCI 151	CSCI 152	CSCI 151	CSCI 152
How many times do you watch the video lectures before class?				
Only one	70%	63%	52%	58%
Two	25%	35%	27%	30%
Three or more	3%	0%	11%	7%
Almost never	3%	2%	9%	4%
Time spent outside of class on this class compared to others				
Much more	31%	13%	49%	28%
A little more	15%	27%	27%	27%
About the same	33%	40%	18%	30%
A little less	15%	15%	4%	13%
A lot less	5%	6%	2%	3%
Preferred method of instruction				
All face-to-face	5%	10%	10%	2%
Mostly face-to-face	13%	13%	7%	9%
Equal	61%	56%	45%	42%
Mostly online	21%	13%	21%	37%
All online	0%	8%	17%	11%

In addition to comparing average final exam scores, we also used online questionnaires in an effort to gauge the students' relative level of involvement in the online components of the courses, and to get their opinions on the use of the flipped classroom. These questionnaires were given during Spring 2015 and Spring 2016 to all students enrolled in CSCI 151 and 152, the results of which can be found above.

The results for the first two questions suggest that most students are indeed spending time outside of class on the course, and are viewing and re-viewing the video lectures. The page-view tracking done by WordPress supports this for Spring 2016, in that most students were opening the individual lesson pages multiple times. The results for "Preferred method of instruction" suggest that students, on average, prefer a balance between face-to-face and online instruction, although there is a slight shift towards online instruction in Spring 2016.

As part of the questionnaires, students were also asked to provide comments on what they would like to stay the same in the course. Out of the 100 respondents to this particular question, 84 mentioned that we should keep the video lectures in some form. Another question that was asked was what should be changed about the course. The answers varied, but only 4 of the 94 respondents suggested that we completely revert back to a traditional lecture-based model of instruction.

VI. Future Work

Given the preliminary results and student feedback, we feel encouraged about our choice in flipping our introductory programming courses. If we can simply maintain the quality of student learning and keep them engaged when we have to deliver these courses to a larger volume of students, we will consider our flipped classroom initiative a success.

As we go forward, we will continue to collect and compare exam scores for these classes, and hopefully get a more definite idea on the impact of flipping on student performance. We also would like to compare student performance between the different majors, and perhaps see if the flipped approach is more effective for different groups. We will also continue to use questionnaires to get student feedback, though we will likely supplement this with data gathered from end-of-semester student course evaluations.

ACKNOWLEDGMENT

The authors would like to thank John Zumbrunnen and Gregory Moses from the University of Wisconsin-Madison for their advice and feedback on using the flipped classroom during the early stages of the project.

REFERENCES

[1] M. Lage, G. Platt and M. Treglia, "Inverting the Classroom: A gateway to Creating an Inclusive Learning Environment," Journal of Economic Education, vol. 31, no. 1, pp. 30-43, Winter 2000.

[2] J. Foertsch, G. Moses, J. Strikwerda, and M. Litzkow, "Reversing the Lecture/Homework Paradigm Using eTEACH Web-based Streaming Video Software," Journal of Engineering Education, vol. 91, issue 3, pp. 267–274, July 2002.

[3] L. Abeysekera and P. Dawson. "Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research," Higher Education Research & Development, vol. 34, no. 1, p. 1-14, 2015.

[4] R. Talbert, "Inverting the Linear Algebra Classroom," PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Studies, vol. 24, issue 5, pp. 361-374, 2014.

[5] G. S. Mason, T. R. Shuman, and K. E. Cook "Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper-division engineering course," IEEE Transactions on Education, vol. 56, no. 4, pp. 430-435 , Nov. 2013.

[6] S. Findlay-Thompson and P. Mombourquette, "Evaluation of a flipped classroom in an undergraduate business course," Business Education and Accreditation, vol. 6, no. 1, pp. 63-71, 2014.

[7] J. D. Tune, M. Sturek, and D. P. Basile, "Flipped classroom model improves graduate student performance in cardiovascular, respiratory, and renal physiology," Advances in Physiology Education, vol. 37, no 4, pp. 316-320, Dec. 2013.

[8] J. L. Bishop and M. A. Verleger, "The flipped classroom: A survey of the research" in 120th ASEE Annual Conference & Exposition, American Society for Engineering Education, June 2013.

[9] M. N. Giannakos, J. Krogstie, and N. Chrisochoides, "Reviewing the flipped classroom research: Reflections for computer science education," in Proceedings of the ACM Computer Science Education Research Conference (CSERC '14), pp. 23–29, 2014.

[10] M. L. Maher, C. Latulipe, H. Lipford, and A. Rorrer. "Flipped Classroom Strategies for CS Education," In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15), pp. 218-223, 2015.

[11] E. F. Gehringer and B. W. Peddycord, III. "The Inverted-Lecture Model: A Case Study in Computer Architecture," In Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13), pp. 489-494, 2013.

[12] H. Jonsson, "Using flipped classroom, peer discussion, and just-in-time teaching to increase learning in a programming course" in Proceedings of 45th Annual IEEE Frontiers in Education Conference (FIE 2015), pp. 1–9, 2015,

[13] G. C. Lee and P. Lee. "Data Structures in Flipped Classroom: Students' Effort and Preference," In Proceedings of the 2015 IEEE Conference on Learning and Teaching in Computing and Engineering (LaTiCE), pp. 152-155, 2015.

[14] Y. Hayashi, Yasuhiro, K. Fukamachi, and H. Komatsugawa. "Collaborative Learning in Computer Programming Courses That Adopted the Flipped Classroom," In Proceedings of the 2015 IEEE Conference on Learning and Teaching in Computing and Engineering (LaTiCE), pp. 209-212, 2015.