THE USE OF A SERIES OF ONLINE MINI-LECTURES TO DELIVER FACTS IN FIRST YEAR PROGRAMMING

Carol Hulls and Chris Rennick
University of Waterloo
chulls@uwaterloo.ca, crennick@uwaterloo.ca

Abstract *- In the first year programming course given to ME and MTE students at uWaterloo, four hours of traditional classroom instruction have been replaced with a series of short online mini-lectures that deliver some of the basic facts necessary to be able to code programs. The students' comprehension of this content is assessed online by quizzes and on the midterm exam. This approach was used in a course which was not otherwise delivered online. The goal was to front-load the course to make space for a design project later in the term. The online mini-lectures were designed to be "lecture-time neutral". The accelerated start of term allowed threshold concepts to show up on assignments a week earlier than with the traditional approach, giving students an additional week of practice with these topics. This led to noticeable gains in understanding on the final exam. Survey data was collected, and focus groups were run, to capture student feedback on the approach; additionally, course grades were analyzed to assess impact on student knowledge of course material.*

Keywords*:* First Year, Internet-Based Instruction, Blended Learning

1. INTRODUCTION

To address concerns of student engagement a high-impact [1] course project [2] was added to the first programming course (GENE 121 – Digital Computation) that students in Mechanical Engineering (ME) and Mechatronics Engineering (MTE) take at the University of Waterloo. This course is offered to MTE students in the first semester of first year, and is offered to ME students in the second semester of first year. One of the challenges in providing a complete design experience in the first year of the curriculum, within these existing courses, is finding ways to devote sufficient time to the design project. In addition, while much of the project is based on "do-learn" [3] [4] as a philosophy, as the design project involves writing software, the students need at least some exposure to, and practice with, basic programming skills before they begin their design.

After the fall 2013 MTE offering of the course, the course teaching team identified two areas which they felt would improve the learning experience for the students, and improve the experience of teaching the course for the instructor:

1. The lecture schedule needed to be made more flexible to simplify the process of scheduling course deliverables and lectures, and provide room for the project to expand, and
2. If possible, functions should be introduced pre-midterm, and assessed on the midterm exam.

Meyer and Land describe a threshold concept "as akin to a portal, opening up a new and previously inaccessible way of thinking about something" [5]. For students to succeed as engineers, they need to be able to take a large problem, break it up into smaller pieces, design appropriate solutions for these smaller pieces to these sub-problems, and integrate these pieces into a working whole. In procedural programming, functions provide this ability, and it is a major watershed moment in the course when students are able to identify, design, and implement their own functions to solve a larger problem. For students to arrive at this awakening, they need early exposure to the concept, and need repeated practice with the mechanics of writing and using functions. Of the threshold concepts in the course (loops, arrays, and functions), functions historically were not taught before the midterm.

To achieve the two goals stated above, the course instructor shifted 4 hours of in-class instruction to a series of short online videos followed by drill and practice quizzes on the video content. It is important to note that the total number of instructional hours did not change. Decoo argued that drill and practice problems incorporated as part of computer-assisted language learning provide greater precision in their evaluation of responses when dealing with clearly defined linguistic elements in mostly unambiguous contexts [6]. In computer programming, the linguistic elements can be considered to be the language syntax and its basic commands. For example, in C++, coding `cout << setprecision(2);` means numbers will be displayed with a precision of 2. This is just an unambiguous fact that students need to know.

Section 2 of this paper describes the full motivation behind implementing these "online mini-lectures", while Section 3 describes the authors' implementation within the overall course. The results, discussed in Section 4, show that these mini-lectures and quizzes have been very well-received by students, and have had no detrimental effect on the overall course grades. In addition, the flexibility the online content provides the course has improved student performance in important, targeted ways, and improved the enjoyment of teaching the course for the instructor. Section 5 contains the conclusions and recommendations of this initiative.

2. MOTIVATION

For the winter 2010 semester, GENE 121 for ME students was changed to remove two assignments and make way for a two week long, open-ended robotics project. This project was expanded for the ME students in 2012 to have a larger emphasis on the design process and on mechanical design. The project was introduced to the MTE students in fall 2011, and involved both programming and mechanical design [2]. Throughout this time, student feedback overwhelmingly pointed to a love of the project, and a wish for more time during the semester to work on it. As the project was implemented primarily as a means to get students to practice the programming skills they had been learning in lectures, and the teaching team did not want to harm the students' chances to succeed in the course, and in their semester overall, there were a number of constraints limiting the expansion of the project. On the front end, enough programming content has to be taught to the students before letting them begin the course project; it is difficult to begin designing a large program, if you don't understand principles such as functions in C++, for example. On the back end of the project timeline, the project could not expand beyond the second last week of semester to ensure the students were prepared for their final exam. Time was also required to allow students to reflect on their project experience by producing a final report.

In addition to the student desire to see the project expand, the first year ME curriculum was being revised for the winter 2015 semester. The new version of GENE 121 for ME students was designed to be a larger course, with a co-instructor teaching mechanical engineering principles, and elements of the design cycle, leading into an expanded project with much more emphasis on engineering design and specifically the mechanical design of the students' projects. This new course, ME 101 – Introduction to Mechanical Engineering Practice 2, has been excluded from all analysis and discussion in the rest of this paper as it shares too few similarities with GENE 121 as it exists for MTE, and as it existed for ME in the past.

In response to the above issues, the teaching team of GENE 121 started brainstorming ways of gaining flexibility in the teaching schedule. One idea to gain flexibility was to explore whether it was possible to teach the same content to the students in a shorter time frame. This would require a means of adding additional lecture hours earlier in the term, so work on the project could begin earlier in the semester. From prior experience, the teaching team had seen an improvement in student performance on their final exam when difficult threshold concepts (such as functions) are introduced to the students before their midterm exam, so it was hoped this accelerated pace could also move at least one of these topics from the second half of the semester to before the midterm.

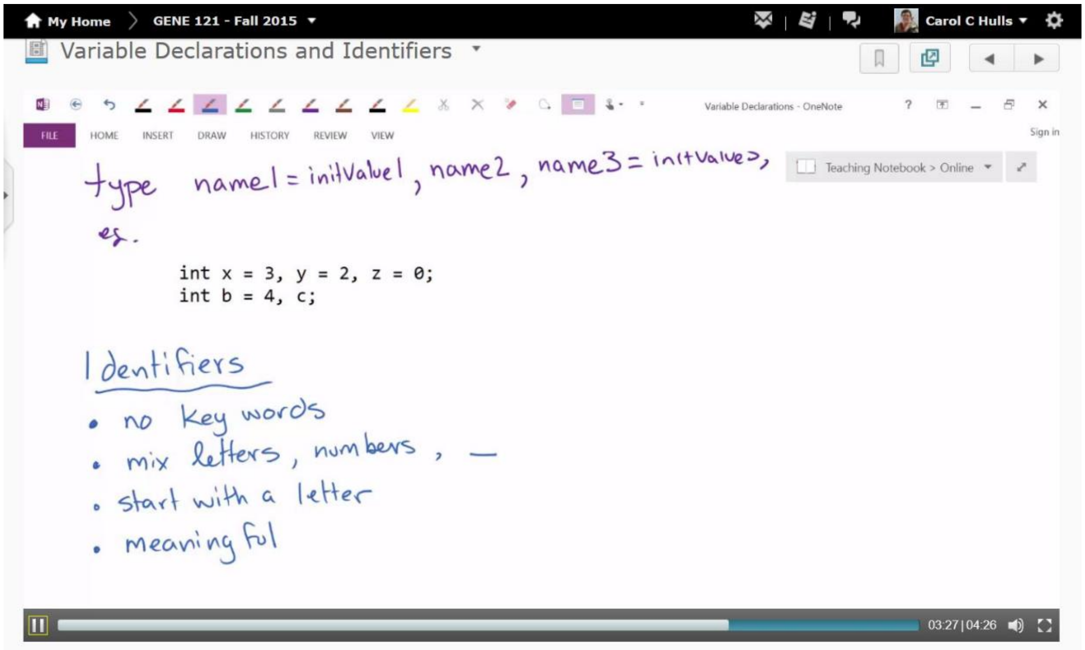
The method chosen to address these concerns was to deliver some of the early, simple, fact-based topics of learning a new programming language using an online platform to allow for an accelerated first month of the semester. Moving a few hours of lecture from in-person in the second half of the semester, to online in the first half of the semester was expected to achieve both goals. However, it was recognized that careful monitoring would be needed as the instructors had concerns about a negative result due to the faster pace.

3. IMPLEMENTATION

3.1 Course Structure

Both instructors used tablet PCs for the in-person lectures with a combination of digital pen input and typed instruction. This approach was used to record the mini-lectures using Camtasia by TechSmith [7], further emphasizing that the information was being delivered as a lecture within the course. All editing of the videos was done in Camtasia. The software provides screen capture abilities and is straightforward to learn. In GENE 121 for ME and MTE students, 4 out of a total of 36 lecture hours are delivered using this approach. For the first four weeks, students are given a set of on-line videos, each video ranging in length from 1 to 15 minutes. A given video covers a single topic so as to enable easy review of that particular topic. Student feedback from the first offering when 50 minute videos were used instead clearly indicated that students prefer the videos to be 5-10 minutes in length and are not concerned about the number of videos that they need to watch to have 50 minutes of material. Figure 1 shows a screen capture from one of the videos showing naming conventions and rules.

After accessing all of the videos for a week, a quiz of 10-20 short answer questions becomes available that is graded and worth 0.5% of their overall course grade. While this incentive is sufficient for most of the students, the remaining students tend to access the material once it starts to be used in the regular lectures and assignments as they realize they have missed some material. Care should be taken on the part of the instructor to not re-teach the online material again in regular lectures. D2L, which is UW's chosen learning management system, was used to implement the quizzes. As the course topics selected for the mini-lectures are ones where there is little to no interpretation required, the quizzes were implemented using multiple choice, multiple select, matching, and fill-in-the-blank. Thus most of the difficulties of dealing with different programming styles or having to parse equivalent



type name1 = initValue1, name2, name3 = initValue3,
eg.

```
int x = 3, y = 2, z = 0;
int b = 4, c;
```

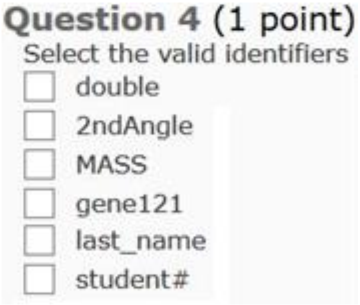
Identifiers

- no key words
- mix letters, numbers, _
- start with a letter
- meaning ful

Figure 1 Screen Capture of Online Mini-Lecture Video

syntax are avoided [8]. Figure 2 shows a typical quiz question that tests students' knowledge of variable naming (that corresponds to the video segment shown in Figure 1).

Students are allowed an unlimited number of quiz attempts. The access logs show that typically they access the videos, attempt a quiz, then review the relevant videos for any questions that were marked as incorrect. Students tend to repeat quizzes 2-5 times, with an average of 3 attempts, before they are satisfied with a perfect or near perfect score.



Question 4 (1 point)

Select the valid identifiers

□ double
□ 2ndAngle
□ MASS
□ gene121
□ last_name
□ student#

Figure 2 Online Quiz Question

During the first four weeks of term, students would attend their regularly scheduled lectures, and then in addition, they were responsible for watching one hour's worth of lecture material as online videos. Later in the term the number of lectures in a given week are reduced, allowing students to more effectively study for their midterms and to effectively manage the demands of their end of term projects. In this way the overall number of instructional hours has not changed (4 hours online + 32 hours in person, vs 36 hours in person in a traditional course).

The University of Waterloo recommends students spend 1 hour working at home on a course for every hour of lecture, tutorial or lab time. This means that students have approximately 5 hours to spend on GENE 121 at home. Inexperienced programmers in the first 3-4 weeks of GENE 121 have historically not needed the full 5 hours at home to stay current with the material, while experienced programmers have needed little or no time at home. The instructors decided that students could handle approximately one hour of additional at-home instruction in each week at the start of term without expanding beyond this 5 hour limit. By the end of week 4, the topics in the course are becoming more complex, especially with the accelerated pace these videos allow for, that identifying topics for the online videos becomes much more difficult. In addition, the students by this point are also typically using closer to the full 5 hours of time at home on course topics, finishing assignments, seeking help, etc.

Using this approach allowed the instructor to shift the delivery of course content approximately one week towards the beginning of the term with respect to the calendar, and got the students programming faster. Decisions and repetition appeared on the weekly assignments one week earlier than they would with a more traditional schedule, for example. Accelerating the pace of topic introduction by a week also allowed one of the key threshold concepts of the course: functions, to be introduced pre-midterm, and therefore tested on the midterm exam. Although not formally studied until the analysis presented in this paper, prior experience had led the instructors to believe that when functions are tested on the midterm, students show a greater understanding of the topic by the final exam. Section 4 of this paper presents an analysis based on student's demonstrated ability to write, and use, functions on final exams.

3.2 Identifying Topics

The students are able to watch the videos at the time of their choosing, so the topics selected for this treatment have to be sufficiently stand-alone that they can be learned at any time during the week they are assigned. The topics for the first month of programming lectures can be classified two ways: those that involve some discussion as to how to best use the knowledge, and those which involve programming facts ("X is coded in the following manner"). For the former types of topics, the students benefit from the classroom interaction that results from having a discussion between students and instructor. For the latter types of topics, the content is unambiguous and an experienced instructor can anticipate any questions that students may have about the material. Learning this material is well-suited for a drill-and-practice approach that allows students to work through the details of the topic. Thus these topics fit well with a model where the topic is introduced as an online video, and then the students complete an online quiz about the topic.

Care must be taken to ensure that the identified topic is not a sub-topic of, or close relative of, a more complex topic. For example, the first time this approach was implemented, reading a value from a file was introduced in an online video. To the instructor, this topic was very mechanistic as it involved only the parsing of data types. However, the topic is closely related to how data from the file should be organized within a data structure in the program, and is tied to how the computer tracks what information has been read from a file. Students were unable to separate the mechanism of reading an input file from the concept of input files, storing large data, etc., and so found the video information confusing. The sub-topic (the mechanism of reading a file) could not be separated from the major topic (input files, large data storage, parsing of data types) and so could not be reduced to easily digestible pieces which could be moved online. It required a great deal of effort on the part of the teaching team to clarify these misunderstandings in the tutorial sessions. In all subsequent offerings, file input was moved back to in-class lectures.

There are additional challenges to using this approach. As students are given one week to watch the lectures and complete the quiz, the topics must be ones that do not have to be delivered in a strict order. For example, students should learn DeMorgan's rule and how it can be used for coding conditions, but this knowledge does not need to be introduced at the same time as decisions are taught as students start by coding with very simple conditions, and so any in-class examples do no use DeMorgan's until a bit later. Yet it is not a problem for DeMorgan's to be taught when conditions are introduced. So there is a window of about a week outside the scheduled lecture time when students can learn this material without disrupting the flow of topics. In an introductory programming course where most of the material builds throughout the term it can be difficult to find topics that have few time dependencies. For reference, the full list of topics taught online in winter/spring 2014, in order, are listed in Table 1 below.

Table 1 Online topics in Winter/Spring 2014 to ME students

Winter 2014	Spring 2014
Declaring variables, console inputs, literals, constants, math operators, type conversion, advanced console input, compound operators, relational & logical operators, file input & output, output formatting, increment & decrement	Types, console inputs, literals, constants, math operators, type conversion, compound operators, output files, relational operators, advanced console input, output formatting, and increment & decrement

4. IMPACT ON STUDENT LEARNING

4.1 Student Opinion

It is interesting to note student attitudes toward the use of the on-line mini-lectures as the term progresses. At the beginning of the term there is usually some grumbling about the amount of work required to watch an extra hour's worth of lecture material. (Although one student did comment that watching the videos goes well with doing laundry.) However, later in the term when they are deeply engrossed in their design projects and have gained an understanding as to the typical workload for an engineering student they come to appreciate that there are fewer scheduled programming lectures.

The authors anonymously surveyed the students in each of the first two offerings of the course with the new online content (winter and spring 2014 semesters). These surveys were delivered in the week before the midterm (pre-Mid), and at the end of term (EOT). In the winter semester, 44 out of 124 students responded to the pre-midterm survey, and 40 out of 124 responded to the end of term survey. In the spring semester, 43 out of 115 students responded to the pre-midterm survey, and 27 out of 115 responded to the end of term survey. The results of these surveys are summarized in Table 2 below. As mentioned in Section 3 above, the only significant difference in the content that was delivered online between these semesters was that file input was moved back into the classroom for the spring term. This change could explain the reduction in students saying the quizzes did not help them on the weekly assignment.

Table 2 Survey responses, ME students in winter/spring 2014

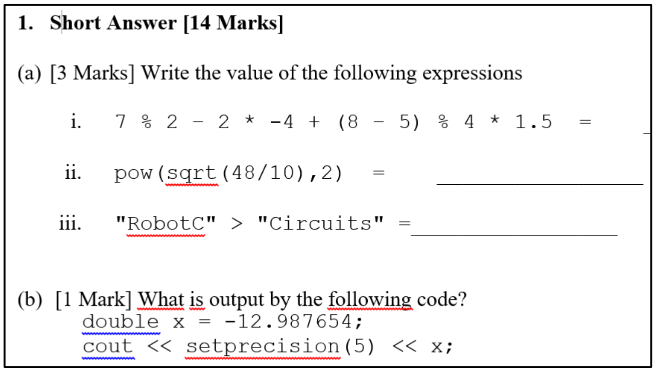
Question	**Winter**		**Spring**	
Content Recognition/Understanding	**Pre-Mid**	**EOT**	**Pre-Mid**	**EOT**
Agreed the online mini-lectures covered content adequately	41(93%)	32(80%)	35(81%)	21(78%)
Agreed they could recognize all topics on the weekly assignment	19(43%)	24(60%)	23(53%)	16(59%)
Agreed they could recognize most topics, with some feeling unfamiliar	20(46%)	12(30%)	13(30%)	11(41%)
Pace of Course				
Felt new topics were coming in a little fast on weekly assignments	12(27%)	12(30%)	16(37%)	9(33%)
Felt new topics were arriving much too quickly	5(11%)	5(13%)	5(12%)	1(4%)
Online Quizzes				
Felt the online quizzes were not helpful	4(9%)	N/A	4(9%)	N/A
Felt the content of the quizzes did not help them on the weekly assignment	15(34%)	N/A	3(7%)	N/A

Late-term course critiques and end of term surveys show that almost all students favour this approach, with students commenting: "[I] really liked the online mini-lectures, they were a great way to learn small specific programming concepts", "the online lecture tutorials helped a lot. There should have been one for every topic", and "The online mini-lectures and quizzes with unlimited trials were a blessing". In the end of term survey given to the Mechanical Engineering students in the winter 2014 term, 40 students responded, of these: 33 (83%) felt that cancelling some lectures at the end of term was either beneficial, or somewhat useful; and 27 (68%) would, or might, prefer one fewer assignment with an extra week to work on the project, which would only be possible with some content delivered online. In the spring 2014 term, 27 students responded, of these: 20 (75%) felt that cancelling some lectures at the end of term was either beneficial, or somewhat useful; and 17 (63%) would, or might, prefer one fewer assignment with an extra week to work on the project.

4.2 Impact on Student Grades

The knowledge taught in the online mini-lectures and online quizzes is directly assessed each term on the first question of the midterm exam (and indirectly elsewhere). This question typically has between 6 and 10 parts to it, made up of short answer questions which test the students understanding of these introductory concepts. These short answer questions typically have students write what the program would output to the screen (including correct output formatting), or have the students evaluate an expression as the computer would. Figure 3 below shows a sample of this style of question from a previous midterm.

The student performance on these questions as well as their overall midterm and final exam averages are summarized in Table 3 and Table 4 below, where an asterisk indicates the terms where the online mini-lectures and quizzes were used.



1. Short Answer [14 Marks]

(a) [3 Marks] Write the value of the following expressions

i. `7 % 2 - 2 * -4 + (8 - 5) % 4 * 1.5 =`

ii. `pow(sqrt(48/10),2) =` ____________________

iii. `"RobotC" > "Circuits" =`____________________

(b) [1 Mark] What is output by the following code?

```
double x = -12.987654;
cout << setprecision(5) << x;
```

Figure 3 Sample short answer midterm exam question

Table 3 Midterm and final exam performance - Mechanical

	Midterm Q1 Avg	Midterm Exam Avg	Final Exam Avg
W 2013	66%	60%	69%
W 2014*	73%	74%	71%
S 2014*	64%	69%	65%

Table 4 Midterm and final exam performance - Mechatronics

	Midterm Q1 Avg	Midterm Exam Avg	Final Exam Avg
F 2012	75%	75%	64%
F 2013	79%	67%	68%
F 2014*	73%	74%	71%
F 2015*	76%	80%	68%

The results show that moving the content out of the traditional classroom did not impact student grades for the particular topics selected for the videos. So from this perspective there is no cost to moving the material online. While the expected improvement due to using drill and practice for review did not occur, this may be due to the fact that students will use this technique while studying for the midterm as copies of the old exams are made available to the students, and so they drill and practice with the material regardless of the delivery format.

All semesters summarized in Table 3 and Table 4 were taught by one of the two authors, with Mr. Rennick teaching the spring 2014 ME students, and Dr. Hulls teaching the other six terms. The amount of teaching team support (TAs, staff) was constant throughout this time period. It is worth mentioning that since the online mini-lectures and quizzes allow the start of the term to be taught at an accelerated pace, additional content can be taught before the midterm exam. The content taught before the midterm exam (that students are then responsible to know for the exam) is summarized in Table 5 and Table 6 below.

Table 5 Topics taught pre-midterm - Mechanical

W 2013	Types, decisions, loops, file I/O, 1D arrays, RobotC, pass by value functions
W 2014	Same as W 2013, plus 2 lectures of additional large in-class examples, and students coded examples during lecture time
S 2014	Same as W 2014

Table 6 Topics taught pre-midterm - Mechatronics

F 2012	Types, decisions, loops, file I/O, 1D arrays
F 2013	Same as 2012, plus RobotC
F 2014	Same as 2013, plus pass by value functions, additional large in-class examples
F 2015	Same as 2014

Due to scheduling differences at the University of Waterloo, the fall semester has the midterm exam one week earlier than in winter and spring, accounting for some of the differences seen between Table 5 and Table 6.

For the re-worked versions of the course, 1-3 entire lectures were used to work through large programming examples involving decisions, loops and arrays with the students, instead of introducing new topics. Working through a program which solves one of these problems takes an entire lecture. These problems are typically solved by the instructor, with class discussion, though these lectures are sometimes taught in large computer labs where the students are programming along with the instructor. This time was available because there was more lecture time before the calendar dictated that the midterm was scheduled and the project needed to start. As the schedule was time neutral for the term, it meant that fewer details of object-oriented programming, such as `struct`s or explicit calls using `(*this)` were covered later in the term. The trade-off is reasonable as it gives students a better grasp of the fundamentals, and the frustration level with their weekly assignments drops as they are better equipped to take on larger problems, albeit at an introductory level, as they were shown a complete design in lectures. This comes at the expense of details that a student should be able to learn on their own once they complete the course.

In many cases, the lecture time gained before the midterm was also used to introduce pass by value functions earlier with respect to the term schedule. Functions are one of the threshold concepts for the course and prior experience has shown that teaching functions pre-midterm leads to large gains in understanding by the final exam. Overall, no additional lecture time was devoted to functions in the semester, it was simply their introduction that was moved forwards. Table 7 below shows the average grade for the question(s) which directly target student ability to use/write functions on the final exam; an asterisk indicates semesters where functions were assessed on the midterm as well as the final exam. The data for spring 2013 is missing from Table 7 as functions on that final exam were not assessed in a stand-alone question, but were instead blended throughout several questions on the final exam, and so comparison is difficult.

Table 7 Final exam performance - question on functions

Mechatronics		Mechanical	
F 2011	71%	W 2012*	75%
F 2012	67%	S 2012	71%
F 2013	74%	W 2013*	78%
F 2014*	83%	W 2014*	76%
F 2015*	77%	S 2014*	87%

The data consistently shows that even though the amount of lecture time devoted to functions is unchanged, student performance on this topic improved by a noticeable amount. The one additional assignment which used functions, combined with student motivation to master a topic for the higher stakes midterm exam increases student proficiency as measured several weeks later on the final exam.

As noted earlier, for both the Mechanical and Mechatronics Engineering students, delivering this early content online, instead of in person has little noticeable effect on their performance on the specific knowledge questions, or overall on their midterm and final exams. The primary motivation to move this content online was to create space for the course project in the second half of the term, and though the course project was present in all terms of GENE 121 described above, it has grown in complexity (with more emphasis on planning, software design, and mechanical design) since the online content has been added to the course. It is also interesting to note that their overall midterm exam performance has stayed consistent (and may be increasing slightly, although more data is needed to determine this) since the online content was introduced to the course, even though additional topics were assessed on the midterm exam.

4.3 Instructor Perspective

The topics selected for the mini-lectures are ones that are straightforward with no interpretation needed. It is hard to make these topics particularly interesting, so they tend to make the overall lecture duller and students are more likely to disengage in the classroom. It then requires greater effort on the part of the instructor to get their attention. Whereas in the terms where the mini-lecture approach was used, the instructors noted an improvement in the overall tone of the lectures. Student expectation is that lectures are a balance between a means of information transmission and a forum for discussion on design decisions with regards to their code, rather than primarily a means of getting facts read to them. Students who are highly engaged in the lecture material and who use the lecture time to ask probing questions are a joy to teach. It is gratifying for an instructor to see so clearly the value of their presence in the classroom rather than feeling that they are just presenting information that is readily available on the internet.

Later in the term when the students are focused on the project, fewer lectures are scheduled for the course. It is only a small amount of time off for the students, but because they are so busy, even one hour of free time seems to make a difference in their stress level. They are being required to absorb less material, and can use that hour productively. It makes a good time for the instructor to schedule one on one meetings with the project teams without having to use the students only free daytime hours at lunch or dinner time.

There is one notable cost to these improvements, which is the additional instructor time that is required. The time to produce the videos can be significant, particularly for the first time until the instructor gets familiar with how to deliver a mini-lecture while recording so that it is easy to edit. For example, since it is much easier to remove then to add content, the authors learned that when mistakes are made, don’t stop recording, just “rewind” the lecture, pause, and resume talking at the earlier point. With this technique, editing just involves cutting a portion of video and audio together. The re-usability of the videos has been somewhat limited in practice; this is due in part to a learning curve on the part of instructors in terms of selecting the most suitable topics. After several implementations of the online content, this factor has reduced greatly. However, there is always some flux in the topics for the videos due to unavoidable differences in lecture schedule, which can vary from term to term based on how courses are scheduled, holidays, etc. In practice this has meant that about half of the lecture videos are re-used from term to term with no changes, and an additional 20 percent need minor editing. The problem may eventually disappear completely once all possible, suitable topics have received this online treatment.

5. CONCLUSION

Implementing these online lectures was purely a means to an end: the course needed the flexibility and space to allow the course project to expand, while staying as time-neutral to students, and the instructor, as possible; and the teaching team desired a way to introduce functions pre-midterm. Moving four hours of lecture online addressed these concerns, and had the added benefit that the increased pace of the first month of the semester allowed all course threshold concepts to appear in lecture, and on assignments, a week earlier than before with respect to the calendar.

This shift of topics forwards allowed for the introduction of functions to occur before the midterm exam, which has led to a clear improvement of student understanding of this crucial topic by the final exam. The flexibility allowed for an expanded course project, which ties more directly to the student’s chosen discipline than before, while staying lecture-time neutral.

From the instructor perspective, scheduling the course was much simpler with added the flexibility in the schedule, and moving the more tedious topics online allowed for a higher percentage of in-class time to discuss the more complicated programming topics, programming style, and software design. Lastly, while the time required to produce the online content was greater than expected, at least (as with the students) the extra work load is over in the first month of the semester.

For other instructors debating similar implementations in their own courses, a great deal of care is needed when choosing the topics to move online. These topics need to be unambiguous, and have little to no room for misinterpretation. Even still, in a new implementation of this strategy, it is likely that mistakes will be made, so careful monitoring of the situation is a must, at least for the first offering. In the first offering of the course with the online lectures, file input was taught online and it was noticed very quickly in tutorial sessions that students did not understand the concept. In our case, clarifying the topic of file input required a great deal of effort on the part of the teaching team, especially as the midterm exam was rapidly approaching. The strategy described in this paper is not recommended for instructors teaching a course for the first time as the instructor needs to anticipate the common questions on the topics which have been moved online. With these warnings in place, the benefits of this strategy have far outweighed the costs, especially beyond the first offering.

Acknowledgements

Thank you to the Samar Mohamed and the Centre for Teaching Excellence at the University of Waterloo for their support throughout this initiative.

References

[1] G. Kuh, High-Impact Educational Practices: What They Are, Who Has Access to Them, and Why They Matter, Washington: AAC&U, 2008.

[2] C. C. W. Hulls, C. Rennick, S. Bedi, M. Robinson and W. Melek, "The Use of an Open-Ended Project to Improve the Student Experience in First Year Programming," in *2015 Canadian Engineering Education Association Conference*, Hamilton, 2015.

[3] E. Guizzo, "The Olin Experiment," *IEEE Spectrum*, pp. 30-36, May 2006.

[4] M. Somerville, D. Anderson, H. Berbeco, J. R. Bourne, J. Crisman, D. Dabby, … Y. Zastavker, "The Olin Curriculum: Thinking Toward the Future," *IEEE Transactions on Education*, vol. 48, no. 1, pp. 198-205, 2005.

[5] J. Meyer and R. Land, "Threshold concepts and troublesome knowledge: linkages to ways of thinking and practicing," Enhancing Teaching-Learning Environments in Undergraduate Courses (ETL), Edinburgh, 2003.

[6] W. Decoo, "In defence of drill and practice in CALL: A reevaluation of fundamental strategies," *Computers & Education*, vol. 23, no. 1-2, pp. 151-158, 1994.

[7] TechSmith, "Camtasia," [Online]. Available: https://www.techsmith.com/camtasia.html. [Accessed 5 May 2016].

[8] C. C. W. Hulls, A. J. Neale, B. N. Komalo, V. Petrov and D. J. Brush, "Interactive Online Tutorial Assistance for a First Programming Course," *IEEE Transactions on Education, Special Issue on Web-based Instruction*, vol. 48, no. 4, pp. 719-728, 2005.