DePaul University
UNIVERSITY LIBRARIES

DePaul University
Via Sapientiae

Technical Reports

College of Computing and Digital Media

8-2008

Preliminary Results with a Targeted Online Java Course

Amber Settle
DePaul

Will Marrero
DePaul

Chad Settle

Recommended Citation

Settle, Amber; Marrero, Will; and Settle, Chad. (2008) Preliminary Results with a Targeted Online Java Course.
https://via.library.depaul.edu/tr/3

Preliminary Results with a Targeted Online Java Course

Amber Settle
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604
(312) 362-5324
asettle@cdm.depaul.edu

Will Marrero
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604
(312) 362-5065
wmarrero@cs.depaul.edu

Chad Settle
University of Tulsa
600 South College
Tulsa, OK 74104
(918) 631-3157
chad-settle@utulsa.edu

ABSTRACT

While the College of Computing and Digital Media has offered online courses for 7 years, courses targeted specifically at online students remain in the minority. In this report, we investigate both student learning and student satisfaction with a targeted online introductory Java course developed by the first co-author. Initial results show that this targeted course has equivalent outcomes with respect to student learning and strongly improved student satisfaction.

1.INTRODUCTION

The College of Computing and Digital Media (CDM) at DePaul University has offered online courses since 2001 using a system called Course Online (COL) to capture the classroom experience and redistribute it to online students. This approach has been successful in making a wide variety of technology courses available to online students and has resulted in large growth in online degree programs. However, CDM faculty have also experimented with other pedagogical approaches to online courses. Alternative discrete mathematics, computer architecture, and electronic commerce courses were created as early as 2002, although none of the alternative online approaches have been as broadly used as the COL model that is standard to CDM.

In this report, we discuss a first experiment with an alternative approach to teaching online Java programming. Materials for the course discussed in this paper were developed in the Fall quarter 2007, and the course was first offered during the Winter quarter 2008. The data mentioned in the sections below are taken from the Winter 2008 offering of the new course and a Winter 2008 offering of the traditional COL model of the course, as well as the Spring 2007 offering of the CO-style course. The new course and the Spring 2007 courses were taught by Dr. Settle and the Winter 2008 COL course was taught by Dr. Marrero.

2. COURSE STRUCTURE

The course considered in this report is CSC 211: Programming in Java I, which is a first-quarter course in Java programming. The course covers the basics of programming with Java, including variables, data types, expressions, standard input and output, using objects and methods from pre-defined classes including the String, Scanner, Math, and Character classes, logical expressions, control statements including if, if-else, switch, while, do-while, and for, file input and output, writing methods, and arrays. The course is taken both by majors and non-majors taking it as a part of the Scientific Inquiry Domain of the Liberal Studies Program. Students in the class include undergraduates and graduate students taking the course in the prerequisite phase of various Master's degree programs in the School of Computing at CDM. Few courses at CDM serve as diverse an audience as CSC 211.

The first two co-authors of this report taught COL versions of the introductory Java course, and Dr. Settle developed and taught a Java class developed specifically for online students. The remainder of this section discusses the structure in each of the online Java courses considered in this report.

2.1 COL courses

When online courses were introduced to CDM in the Spring quarter 2001, the goal was to provide a large variety of distance-learning courses while minimizing the impact on the faculty and the regular sections of courses. To do so, staff at CDM developed a system called Course Online (COL). COL simultaneously captures audio, video, the instructor's notes written on the whiteboard, and the images displayed on the instructor's computer screen. The capture of the information is done automatically, and although the equipment is monitored remotely, there are no staff in the classroom when the recording is done. The various streams are then synchronized, and by the morning after the lecture they are made available to students registered in the class. The recordings remain online for the entire quarter and are a part of an integrated course management system that allows faculty to post course information such as the syllabus, assignments, class notes, and grades and includes a homework submission system. More technology intensive than labor intensive, the COL approach is unique [1]. Though the system is asynchronous, online students can hear the comments and questions made by in-class students. The online students and the students in the regular section also can communicate easily with each other via threaded discussion boards and with the instructor via e-mail and posted announcements.

COL is used to record a majority of classes taught at CDM, including all evening classes. Each of the Masters degrees at CDM have a prerequisite phase designed to prepare students for graduate study in their chosen area, a common solution for students who wish to switch areas of study after completing their undergraduate degree [5]. CDM evening classes enable Masters students who work fulltime to complete their degree programs. Courses that are required in the prerequisite phase of CDM Masters programs are typically assigned a COL section each quarter that they are offered, and courses that are common to a number of Masters degrees are taught every quarter. CSC 211 is required in seven Masters degree programs, and a COL section of the course is available every quarter.

The remainder of this section discusses the COL Java courses taught by Drs Settle and Marrero.

2.1.1 Basic course structure

CSC 211 is an introduction to programming in Java and problem solving. Although students use pre-defined classes, they do not write their own classes so that CSC 211 has a more procedural approach than many introductory Java courses. This approach allows extra time to be spent discussing how to solve problems using algorithms and how to implement algorithms in Java.

The recommended week-by-week schedule for the course is given below and includes references in the textbook for the course [3] and examples of the type of problems students should understand:

Week	Topics	References
1	Writing, compiling, and running Java programs. Problem solving, algorithms, and pseudocode. Printing to standard output.	Chapter 1. Example problem: Calculate a salesperson's monthly pay
2	Identifiers, variables, primitive data types, arithmetic expressions and operators, type conversions. The classes String and Scanner. The Java API and importing packages.	Chapter 2. Example problems: Convert feet and inches into centimeters; Make change by converting cents into quarters, dimes, nickels, and pennies.
3	Calling object methods. Methods in the String class. Input/output with JOptionPane. File input/output with Scanner.	Chapter 3. Example problem: Calculate a student grade.
4	If and if-else statements. Relational operators and Boolean operators. Nested if-else. Switch.	Chapter 4. Example problem: Calculate a cable bill.
5	Midterm exam	None
6	While loops and for loops.	Chapter 5. Example problems: Hi-Lo game to guess a number between 1 and 100; Compute the nth Fibonacci number.
7	Do-while loops. Nested loops. The Math and Character classes.	Chapters 5 and 7.
8	Writing methods: method headers, return types, and parameters. Scope and method overloading.	Chapter 7. Example problem: Write a method to decide whether a String is a palindrome.
9	Arrays: declaring, creating, and initializing. Arrays as parameters. Searching arrays. Two-dimensional arrays.	Chapter 9. Example problem: Find the largest number in an array; Print an array in reverse order.
10	Review or optional topics	None
11	Final exam	None

All CSC 211 instructors are encouraged to use OWL. OWL is an online assessment system for programming classes [4]. The problems are selected from the pre-written questions in the OWL system and consist of matching, multiple-choice, fill-in-the-blank, and short-answer questions as well as short programming questions. The problems are automatically graded by the system, providing the students with immediate feedback including suggestions on improving their answers. OWL is accessed by students using a purchased code, and the DePaul University bookstore sells bundles including the course textbook and an OWL code. OWL codes can also be purchased without a book both from the bookstore and online.

While the above is the recommended approach to CSC 211, each instructor may modify the order of material or include different optional topics. The creation of assessments and the grading policies for each course are also subject to instructor discretion. The sections below discuss the differences and similarities in the approach taken by the first two co-authors.

2.1.2 Differences

One significant difference between the two online Java courses was the mix of assessments in the course and the policies regarding late work. Dr. Marrero's class included OWL exercise sets, programming assignments, a midterm, and a final exam. No late assessments of any form were accepted. The lowest programming assignment score was dropped in the calculation of the course grade. Dr. Marrero also included weekly ungraded lab activities in his course. Dr. Settle's class included lab activities, OWL exercise sets, programming assignments, a midterm and a final exam. Lab activities were completed in class by traditional students and at a setting of their choice by the online students. No late labs or OWL exercise sets were accepted. Programming assignments were accepted up to 4 days after the due date, with a 20% penalty per day. There were 4 lab activities and the lowest score was dropped. The lab activities involved completing short programs or methods and then providing code to test those methods. There were weekly programming assignments and OWL exercise sets, and the lowest score for each was dropped in the calculation of the grade.

The topics covered in Dr. Settle's class varied slightly from the indicated weekly schedule. While and do-while loops were covered just before the midterm exam, although they were not included in the material assessed on the midterm. Writing methods was discussed over the course of two weeks, namely the 7th and 8th weeks. And because of a missing class due to the Memorial Day holiday, two-dimensional arrays were not covered in the course.

Finally, the textbook used in the course changed from the 2nd edition to the 3rd edition in the Summer 2007. Dr. Settle's Spring 2007 COL course used the 2nd edition of the textbook [2], whereas Dr. Marrero's Winter 2008 COL course used the 3rd edition of the textbook [Malik3]. Note, however, that the differences between the two textbooks are minor and do not have a significant impact on the material covered in the course.

2.1.3 Similarities

Despite using slightly different assessments in each course, both instructors assigned similar weights to certain types of assessments. The grade in Dr. Marrero's class was based 20% on programming assignments, 20% on OWL exercise sets, 30% on the midterm exam, and 30% on the final exam. The grade in Dr. Settle's class was based 5% on lab activities, 5% on OWL exercises, 30% on programming assignments, 30% on the midterm exam, and 30% on the final exam. Thus both instructors assigned between 35 – 40% of the course grade to at-home programming problems. Both instructors placed significant emphasis on the importance of completing all exercises and programs in a timely fashion and indicated to the students that successful completion of homework was crucial for understanding the material. Both instructors also emphasized proctored exams in the course, assigning 60% of the course grade to the combination of the exams.

2.2 Online-only course

The alternative online course covers the same material as the COL courses, but the topics in the course are divided into eight distinct "modules". Each module has an associated set of recordings and a set of assessments. Modules are grouped together for online quizzes, and the final exam was comprehensive across all modules. Each module depends only on previous modules, although the assessments emphasize the material that is new to the module.

The course is structured for an 11-week quarter, and the table below lists the week-by-week topic schedule for the course, including the division of material into modules:

Week	Module	Topics	Assessment due
1	M1: Java basics	Running a Java program; Variables, data types, and expressions	None
2	M2: Standard input and output	Using the Scanner and JOptionPane classes	Module 1 assessments
3	M3: Pre-defined classes	Using objects from pre-defined classes, including String, Character, and Math	Module 2 assessments
4	M4: Logical expressions and branching	Logical expressions (Booleans, &&, ǁ, !) and branching statements (if, if-else, switch)	Module 3 assessments; Quiz 1 (Modules 1 – 3)
5	M5: Looping	Looping statements (while, for, do-while)	Module 4 assessments
6	M6: File input and output	Using the Scanner and PrintWriter classes; throwing FileNotFoundExceptions; closing objects	Module 5 assessments
7	M7: Writing methods	Value-returning and void methods; Method headers, return types, modifiers, formal and actual parameters; Method overloading	Module 6 assessments
8	M8: Arrays	Declaring and allocating arrays; Initializing arrays; Accessing arrays; Array algorithms (finding the smallest, finding the largest, finding a specific element, etc.); Arrays of objects; Arrays as parameters to methods; Two-dimensional arrays	Module 7 assessments
9	None	None	Module 8 assessments
10	None	None	Quiz 3 (Modules 7 – 8)
11	None	None	Final exam

2.2.1 Recordings

Recordings for the online-only course focus on example code. There are no "lectures" in the traditional sense of the word. Instead, students are provided with written lecture notes that summarize the material covered in each module and a reading assignment in the textbook for the course [3]. At various points in the lecture notes, students are referred to relevant examples illustrating the syntax and concepts covered in the notes. The recordings discuss and develop the example code in the written lecture notes, highlighting how the code illustrates the important concepts and modifying the program(s) to demonstrate additional concepts and syntax covered in the module.

There were 31 recordings made for the Winter 2008 course. The number of recordings per module and the length of the recordings varied. The early modules had more recordings, with a maximum of 7 recordings for each of modules 3 and 4, and the later modules had fewer recordings, with a minimum of 1 for module 6. The recordings for the later modules tended to be longer, although no recording was longer than 1 hour. The recordings were made using Camtasia and consisted of a view of the development environment, in this case TextPad, and a recording of the instructor's voice discussing the code. The recordings were posted on the COL course management site along with the files for the code. When the code was significantly modified during the recording both the original code and the final version of the code were posted.

A second set of recordings was made each week during the quarter. These recordings provided students with feedback about the assessments that had been submitted the previous week. The recordings again discussed example code, but this code was created by the instructor in response to the assessment submissions. In addition to discussing solutions to elements of the assessment that students found challenging, the recordings provided dos and donts for future assignments. The feedback recordings were 10 – 30 minutes in length and were posted on the COL course management system several days prior to the submission of the next assessments.

There were two additional recordings made prior to the start of the quarter. The first recording was an introduction to the instructor, giving information about her teaching and research interests. The second recording was an introduction to the course and the COL course management system. Both recordings were less than 20 minutes in length and were created to mimic the introduction and orientation to the class that is standard in the first session in traditional classes.

2.2.2 Assessments

A number of different assessments were used in the online-only course to ensure that students had ample opportunity for feedback on their Java programming skills. Each module required the completion of participation exercises, the submission of short answer questions to the OWL online tutoring system, and a longer programming assignment. Throughout the quarter there were 3 online quizzes, also taken using the OWL system. Finally there was an in-person, comprehensive final exam taken during the last week of the quarter.

During each week the first exercises that students are required to complete were the OWL problems. In the early modules the students are given a fixed number of tries to complete the problems, typically 3, a practice necessitated by the matching and multiple-choice questions found in those problem sets. Problems in later modules, particularly questions involving completing or writing small programs, were given unlimited tries for completion. Although the system keeps track of the number of tries each student requires to complete a problem, that information was not used in determining the student's grade. All submissions completed correctly by the deadline were given full credit. Late submissions were not permitted except in extraordinary circumstances. OWL exercises were worth 15% of the course grade.

Once students complete the OWL exercises, they are required to complete the associated participation exercise. To do this, they log into the COL course management system and access the unique discussion forum created for the OWL and participation exercises. There the students post the OWL exercise they found to be the most difficult and the reason they found the exercise particularly challenging. Students are warned not to give away the answer when making their post since they may complete the OWL exercises before other students are done. The participation assignments were originally created to provide the instructor with feedback about the OWL exercises, giving an early way to see what students understood and what was causing them more problems. An unforeseen consequence of the exercises was the benefit the postings provided to students completing the OWL exercises, even without posted answers. Many students commented that they had begun to read the forum prior to completing the OWL problem sets in order to get tips about which exercises required special attention. The participation exercises were worth 5% of the course grade.

The final assessment of each module is a programming assignment. Students are provided with problem descriptions and asked to create a program to solve the specified problem. Early programs are highly scripted, indicating every step that the students need to complete as well as providing sample input and output. Programming assignments for later modules are more open-ended, still providing sample input and output but lacking details about how the program should be completed. Overall, however, programming assignments remained fairly clearly written with specific details regarding syntactic requirements. For example, in the writing methods module students were given the header of the methods they were to write in addition to sample input and output for the overall program. Students were encouraged to add methods or use approaches not seen in the examples or lecture notes, but the assignments did not require it in order to earn full credit. The programming assignments were 30% of the course grade.

At three points during the quarter students are required to take timed, online quizzes. The quizzes were written and administered using the OWL system [4]. The quizzes are open book and open notes, but students were told not to work together and had to sign academic integrity pledges indicating that they understood this requirement. However, the quizzes are not proctored. Quizzes were made available to the students in a fixed time frame, typically 3 days in length. Once the students log into the OWL system and select the quiz, they have a short period of time in which to complete the quiz, ranging from 1 hour to 3 hours depending on the difficulty of the modules covered. The first quiz was given after Modules 1, 2, and 3, the second quiz after Modules 4, 5, and 6, and the third quiz after Modules 7 and 8. Although the structure of the course necessitates cumulative assessments, the quizzes emphasized the material in the modules immediately preceding the quiz. Since the quizzes are written by the instructor, they are not automatically graded and thus students are not provided with instant feedback. Instead the students complete a short program for each question on the quiz and submit their code using the OWL system. The main benefit of using the OWL system is the ability to create short-answer questions that use a (mostly) reliable timing mechanism. Timed quizzes also give the students a better sense of whether they have digested the material than an open-ended assignment due over the course of a week. Feedback and scores for the quizzes were posted both on the OWL system and on the COL course management system.

The only proctored assessment of the course is a cumulative final exam. The final exam consists entirely of completion questions requiring the student to write significant portions of short programs. Partial code, sample input and output, and explicit instructions for each problem are provided on the exam. Students register for the final exam using the COL course management system and are required to take the exam at either a DePaul campus or another approved testing center. The students are allowed to use a fixed number of sheets of notes on the final exam, although the exam is closed book. The final exam must be taken during a 5-day window in the quarter.

The total grade assigned to exams in the online course is 50%. Any student earning an equivalent grade or higher on the final exam than on the average of the quizzes had 50% of his/her grade determined by the final exam. Any student earning a lower grade on the final exam than on the online quizzes had 15% of his/her grade determined by the average of the online quizzes and 35% of his/her grade determined by the final exam. The policy is designed to reward those students who used the quizzes as a way to assess their understanding of the material in the course and improve that understanding.

2.2.3 Support Materials

There were a number of special web pages designed to serve as tutorials for students getting started in the online class. Several pages discussed how to install the development environment, including pages on installing Java, installing TextPad, and running a program using TextPad. Other tutorials discussed how to get started with the COL course management system, including pages on posting a message to a discussion forum, submitting programming assignments, and viewing grades and comments. There was also a page that discussed how to get started with the OWL system.

There were also a number of external links posted to the COL site with relevant information. These included a link to a site about how to use a Java development environment on Macs, the Java 2 Platform SE 6 API, and DePaul related information such as the Academic Integrity Policy and the CDM tutoring page.

Students were also encouraged to remain in frequent e-mail contact with the instructor. The instructor also created a Skype account and publicized the ID on the course syllabus.

3. RESULTS

During the Winter 2008 quarter, there were 12 students who completed the online Java section with Dr. Marrero, and 20 who completed the online section with Dr. Settle. Throughout the quarter, a variety of information about each of the online courses was gathered. This included anonymous course evaluations, student surveys about their experience with online courses in general and the Java course in particular, and assessment data from common final exam questions. Unfortunately, there was insufficient response to the student surveys to draw any conclusions from them, as only 7 students from Dr. Marrero's class and 4 students from Dr. Settle's class completed them. As a result, that data is not considered in this report. Further, there were not enough data points to run a regression on the other data sets. Instead, summary statistics were compiled for the common final exam questions and course evaluations, and in the sections below we discuss the statistics. We also compare course evaluation summary statistics from the Spring 2007 COL section in which there were 12 students to the Winter 2008 COL course with 20 students (but only 19 students completing evaluations), both taught by Dr. Settle.

3.1 Student performance

As a part of another project at CDM, two common final exam questions were devised for each section of CSC 211 taught during the Winter 2008 quarter. The first of these questions asked students to read a String using the Scanner class, store it into a variable, then invoke a method the students had not previously used in order to create a modified String, and finally to display the modified String to the user. The second question required the students to write a method that takes an array as a parameter, checks for the presence of at least one pair of adjacent equal integers, and returns true if there is at least one such pair and false otherwise. The question also asked for a set of three test cases that could be used to verify the behavior of the method. Using methods from a predefined class and writing methods that manipulate arrays were fundamental to the course which is why these questions were selected.

A rubric for the questions was also developed. Each problem was subdivided into individual tasks, and students were rated on each task. Students who fully completed a subtask were ranked with a 4, students who almost fully completed a subtask were given a 3, students who satisfactorily completed a subtask were given a 2, students who poorly completed a subtask were given a 1, and students who did not complete a subtask were given a 0. The table below gives the details for each subtask of the two problems:

Problem 1	
1.1.a	Declare a variable s1 of type String
1.1.b	Identify an appropriate method of the Scanner class to read input from the user

1.1.c	Call an appropriate method of the Scanner class to read input from the user
1.1.d	Assign the inputted String to the variable s1
1.2.a	Declare a variable s2 of type String
1.2.b	Invoke the replace method associated with s1
1.2.c	Use the correct signature of the replace method
1.2.d	Assign value returned by replace to s2
1.3.a	Invoke the correct method to output the new String
1.3.b	Use the correct parameters for the output method
Problem 2	
2.1.a&b	Identify and code a loop to parse the array
2.1.c&d	Identify and code a conditional to test for equality of adjacent elements
2.2.a	Propose three arrays of length 5 as test cases
2.2.b	First array is a meaningful test case
2.2.c	Second array is a meaningful test case
2.2.d	Third array is a meaningful test case
2.3.a&b	Meaningful rationale for the first test case
2.3.c&d	Meaningful rationale for the second test case
2.3.e&f	Meaningful rationale for the third test case

To understand how the student learning in the online-only course compares to the COL course, we compare the average of student performance on these two questions. Here we compare only the Winter 2008 COL course taught by Dr. Marrero and the Winter 2008 online-only course taught by Dr. Settle since no assessment information is available for the Spring 2007 COL course. The table below contains averages for student performance for each task for the first problem in each of the Winter 2008 online Java courses:

Task	Marrero	Settle
1.1.a	3.33	4
1.1.b	3.42	4
1.1.c	3.42	4
1.1.d	3.67	4
1.2.a	3	3.05
1.2.b	2.83	2
1.2.c	3.17	1.9
1.2.d	2.5	2.85
1.3.a	3.5	3.25
1.3.b	2.83	3.25

The table below contains averages for student performance for each task for the second problem:

Task	Marrero	Settle
2.1.a&b	3.08	3.6
2.1.c&d	3.21	2.3
2.2.a	3.42	3.2
2.2.b	3.67	3.25
2.2.c	3.33	3.1
2.2.d	3.33	2.95
2.3.a&b	3.25	3.25
2.3.c&d	3.33	3.2
2.3.e&f	3.17	2.95

Students in the two classes performed similarly (within 0.3 points) on a number of tasks. On the first problem, this included declaring a second variable of type String and calling the correct method to display the modified String. On the second problem, this included proposing three arrays of integers of length 5, giving meaningful test cases for at least one of the arrays, and providing rationales for why the selected test cases were meaningful.

Students in the online-only course performed better on some tasks. On the first problem, this included declaring a variable to hold the String read in from the user, identifying an appropriate method of Scanner to read the String, calling the appropriate method of the Scanner class, assigning the String to the variable, assigning the result of the method call to the declared variable, and passing the correct parameter to the method called to display the String to the user. On the second problem, this included identifying the need to code an appropriate loop to parse the array and then coding that loop.

On the other tasks, the students in the COL course performed better. On the first problem, this included calling the previously unseen method of the String class and using the correct signature of the method. On the second problem, this included identifying the need to code an appropriate conditional to check for equality of adjacent elements and then successfully coding that conditional, and providing multiple meaningful test cases for the method.

The preliminary results seem to show that the students in the online-only course are learning at least as much as the students in the COL course. The result that the students in the COL course show a greater ability to write conditionals and use unknown methods suggests that students in the online-only course need to be given more opportunities to use methods not covered in the lecture notes or examples. Students also appear to need more practice in writing conditionals when dealing with arrays, which may be because arrays are the final module in the course. Providing students with a practice problem set for the last quiz and the final exam may motivate them to study arrays in more depth. It is encouraging that the online-only students are better able to write loops since this was a need that was identified in a recent change to an objects-later approach to the course. We would like to confirm these results by running a regression when more data are available.

3.2 Student evaluations

Every quarter in every class at CDM online student evaluations are conducted. The evaluations are conducted online, and the students must log into a secure system and may submit only one evaluation per CDM course in which they are enrolled. No identifying information about the student is associated with the evaluation, making them anonymous. Completing an evaluation is mandatory for all students enrolled in CDM courses. Course evaluations are completed during the 8th and 9th weeks of the 10 week quarter, although results are not made available to instructors until after grades are submitted. The evaluations consist of 27 multiple choice questions and several sections for comments. The multiple choice questions ask the student to rate various aspects of the course and the instructor for the course. Five of the questions are university-wide, and those questions are rated on a scale of 1 to 5. The remaining 22 questions are on a scale from 0 to 10. For both sets of questions the meaning of a rating depends on the question. In general, a higher number indicates a greater degree of student satisfaction with the area addressed by the question. For the 22 CDM-specific questions, a zero indicates that the student feels the question is not applicable.

The five university-wide questions (labeled Q1 to Q5) are:

1. Given your experience at DePaul, rate the instructor's overall teaching effectiveness
2. The instructor stimulated interest in the subject

3. Given your experience at DePaul, rate the overall quality of the course
4. Overall, this course increased my knowledge or skills
5. I found this course to be [possible answers refer to the level of challenge of the course]

The 22 CDM-specific questions are divided into 10 questions that are course specific and 12 questions that are instructor specific. The course specific questions (labeled QC1 to QC10) are:

1. Was the course well organized?
2. Do you feel the course objectives were accomplished?
3. The amount of work you performed outside of this course was [answers refer to the quantity of work]
4. How difficult was the course material?
5. The textbook for the course was
6. Supplementary reading for this course was
7. The assignments for this course was [answers refer to the fairness and clarity of the assignments]
8. The grading of the homework and exams in this course was [answers refer to the fairness]
9. What is your overall estimate of this course?
10. How valuable was this course in terms of your technical development?

The 12 instructor-related CDM-specific questions (labeled QI1 to QI12) are:

1. How would you characterize the instructors [sic] knowledge of the subject?
2. How would you characterize the instructor's ability to present and explain the material?
3. Does the instructor motivate student interest in the subject?
4. How well does the instructor relate the course material to other fields?
5. Did the instructor encourage participation from the students?
6. Was the instructor accessible outside of class?
7. What was the instructors [sic] attitude? How did they treat you? [answers refer to fairness]
8. How well did the instructor conduct, plan and organize classes?
9. Were the instructors [sic] teaching methods effective?
10. Would you recommend this course to another student?
11. Would you take this instructor for another course?
12. Rate the teaching effectiveness of this instructor as compared to other faculty in this school.

In this report we compare the course evaluations between the Spring 2007 COL course and the Winter 2008 online-only course. These two courses were taught by the same instructor, which allows for more accurate comparisons of student satisfaction with the two approaches to online Java courses. Overall, course evaluations were higher on almost all questions for the online-only course taught in Winter 2008. The table below show average course evaluation ratings for each section:

Question	Spring 2007 COL	Winter 2008 online only
Q1	4.25	4.53
Q2	4.17	4.47
Q3	4.17	4.47
Q4	4.0	4.21
Q5	3.67	3.89
QC1	8.67	9.37
QC2	8.67	9.21
QC3	7.08	6.37
QC4	7.25	7.26
QC5	7.42	6.79
QC6	7.0	7.47
QC7	8.08	8.95
QC8	8.08	9.21
QC9	8.42	8.63
QC10	8.33	8.58

QI1	8.58	9.74
QI2	8.75	9.53
QI3	8.42	8.94
QI4	7.75	7.75
QI5	8.09	9.06
QI6	8.7	8.95
QI7	8.45	9.11
QI8	8.67	9.61
QI9	8.5	9.37
QI10	8.42	9.11
QI11	8.58	9.63
QI12	8.42	9.35

The only questions which were higher for the Spring 2007 COL course were the third and fifth questions in the course-related CDM-specific questions. The third question addresses the amount of work done outside of the course and a higher answer means that the students feel that they were doing more work. From this we conclude that the Spring 2007 COL students felt that the amount of work they were completing for the course was higher than the Winter 2008 students. The fifth course-related question has to do with the textbook for the course. A higher number indicates that the textbook was perceived to be more appropriate and helpful by the Spring 2007 COL students. As noted earlier the textbook was changed after the Spring 2007 quarter, and it appears from this result that the students are less satisfied with the newer edition.

There were also two questions that received equal rankings in both the Spring 2007 COL and Winter 2008 online-only courses. The fourth course-related CDM-specific question which concerns difficulty of the course material, and the fourth instructor-related CDM-specific question which concerns how well the instructor relates course material to other fields, received equal rankings. This result would seem to confirm that the students have an equivalent experience with regards to the Java learned in the course and the way that Java is related to the other CDM courses.

It appears from these preliminary results that the students are more satisfied with the online-only Java course. Again, we would like to confirm these results by running a regression when more data are available.

4. FUTURE WORK

The online-only Java course will be taught again the Fall 2008. The materials for the course will be modified in order to better emphasize the important of writing methods to manipulate arrays and invoking previously unknown methods. Both of these changes involve modifying study aids, OWL exercises, and assignments for the course which is a relatively simple issue. Further study of the resulting assessment information and course evaluations can then be done to ensure that the targeted tasks are improved while not harming performance on the other tasks or student satisfaction with the course.

We would also like to gather sufficient data to perform regressions for both the assessments and the course evaluations. To perform a complete analysis, we need more data from COL and online-only courses taught by Dr. Settle and from COL courses taught by Dr. Marrero. The authors are actively working with the CDM administration to make this possible.

5. ACKNOWLEDGMENTS

We are indebted to Gian Mario Besana for his assistance in developing the assessment questions used to evaluate student performance during the Winter 2008 quarter.

6. REFERENCES

[1] Knight, L. , Steinbach, T., and White J., "An Alternative Approach to Web-based Education: Technology-intensive, Not Labor-Intensive". *Proceedings of the Information Systems Education Conference*, 2002.

[2] Malik, D.S. 2006. *Java Programming: From Problem Analysis to Program Design, Second Edition*. Thomson Course Technology.

[3] Malik, D.S. 2008. *Java Programming: From Problem Analysis to Program Design, 3rd Edition*. Thomson Course Technology.

[4] Online Web-based Learning (OWL), University of Massachusetts at Amherst and Cengage Learning, http://owl.course.com.

[5] Wyatt, R., Milito, E., Epstein, R., and Kline, R., A Graduate Master's Prerequisite Program. *Journal of Computing Sciences in Colleges*, 17, 3, (February 2002), 169-177.