Iteration Visualization for Novice Learners

Jieun Chon

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Application

Clifford A. Shaffer, Chair

Dennis G. Kafura, Co-chair

Eli Tilevich

May 16, 2019

Blacksburg, Virginia

Keywords: Iteration, Visualization, OpenDSA, BlockPy, Computatinal Thinking,

Electronic Textbook

Iteration Visualization for Novice Learners

Jieun Chon

(ABSTRACT)

Iteration, the repetition of computational steps, is a core concept in programming. Students usually learn about iteration in an entry-level Computer Science class. Virginia Tech's Computational Thinking (CT) course is designed to teach non-CS majors computing skills and new ways of thinking. The course covers iteration on Day 8 of the class. We conducted a pretest before, and three post-tests after, Day 8 of the Computational Thinking class in Spring 2018 on 137 students. The pre-test was intended to measure knowledge of iteration before the material was covered. We found from the post-tests that students' knowledge of iteration did not satisfy the course objectives in Spring 2018. In particular, the knowledge gain shown between pre-test and post-tests was not significant. We developed interactive visualizations and exercises for Fall 2018 and Spring 2019. We conducted tests and compared the data from Fall 2018 and Spring 2019 (the treatment) against Spring 2018 (the control). We found that Spring 2019 students had greater knowledge gains than Spring 2018 students. Also, we conducted student surveys in Fall 2018 and Spring 2019 to learn more about their opinions on recall, helpfulness, and reuse of the interactive visualizations. Finally, we analyzed data from the interactive exercises and page use to investigate students' usage behavior.

Iteration Visualization for Novice Learners

Jieun Chon

(GENERAL AUDIENCE ABSTRACT)

Iteration is a process of repeating a set of instructions or structures. An iterative process repeats until a condition is met or a specified number of repetitions is completed. Students usually learn about iteration in an entry-level Computer Science class. Virginia Tech's Computational Thinking (CT) course is designed to teach non-CS majors computing skills and new ways of thinking. The course covers iteration on Day 8 of the class. We conducted a pretest before, and three post-tests after, Day 8 of the Computational Thinking class in Spring 2018 on 137 students. The pre-test was intended to measure knowledge of iteration before the material was covered. We found from the post-tests that students' knowledge of iteration did not satisfy the course objectives in Spring 2018. In particular, the knowledge gain shown between pre-test and post-tests was not significant. We developed interactive visualizations and exercises that were used during Fall 2018 and Spring 2019. We conducted tests and compared the data from Fall 2018 and Spring 2019 (the treatment) against Spring 2018 (the control). To see if there was a statistically significant difference between the absolute score means of three groups, we used independent sample t-tests. Also we used paired sample t-tests to see if there was a greater knowledge gain after using our invention. By analyzing the results of the t-tests, we found that Spring 2019 students had greater knowledge gains than Spring 2018 students. Also, we conducted student surveys in Fall 2018 and Spring 2019 to learn more about their opinions on recall, helpfulness, and reuse of the interactive visualizations. We analyzed data from the interactive exercises and page use to investigate students' usage behavior.

Acknowledgments

This project would not have been possible without the support and help of many people. There are several individuals who deserve special thanks. First, Dr. Shaffer has always respected my identity and guided me not only for research and writing but also made a turning point in my life. Thanks to Dr. Kafura for serving as co-advisor, setting up and delivering all the materials of the CT class and guiding me throughout the research. Also, thanks to Dr. Tilevich for guiding me at the beginning of my thesis writing in the Research Methods class and serving on my thesis committee. I would like to extend my thanks to Cory Bart and Javier Tibau for incorporating my Visualization into their classes. Thank you to all students of the Computational Thinking class in Spring 2018, Fall 2018, and Spring 2019, who became users of my interactive visualizations and exercises, and gave me great feedback. Claire Childress, who really cares for students, helped me from the beginning of my career preparation, and I know I could not be here without you. Also, thanks to Sharon Kinder-Potter, not only for her help as a Graduate Program Coordinator but also for all the support. Thank you Dr. Bob Edmison for working on aggregating Canvas data for this research. I would like to also thank people in our lab: Ehsan, Ayaan, Mostafa, Hamza, Thomas, Tyler, Ri, and Jackson who always discussed, helped and advised me throughout my thesis program and brighten the lab's atmosphere all the time. Thank you to Mary Rose Lunde, Jessica Deters, Evan Massey, Atinut Inthajak, and Virginia Tech Writing Center for helping me on the revising process for this Thesis. Thanks to Dr. Thomas Williams for giving me guiance on the statistical analysis process. I would also like to show my appreciation to the National Science Foundation who partially supported my work through grants DUE-1624320 and DUE-1444094, and Virginia Tech for providing me a Graduate

Research Assistantship and a Graduate Teaching Assistantship. For the last, thanks to my family for their support who built me up.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Introductory students are known to have difficulty learning iteration when using traditional instructional techniques [10]. Instructors often draw tables of variables for each state to explain how each line of the iterative code affects the program's variables [9][29][4]. Despite these varied practices, students still have a hard time understanding the concept. In this chapter, we explain several problems with the previous teaching materials used in a Computational Thinking (CT) course at Virginia Tech. We then propose a solution, interactive visualization, to address this shortcoming. The main contributions of this thesis can be divided into four sections: designing and implementing interactive visualization for iteration, conducting pre-post tests in a classroom-based study, analyzing pre-post tests and usage data to identify the impact of the interactive visualization on student learning, and collecting users' experiences by conducting surveys.

1.1 Iteration in Computer Science Education

Using visualization in Computer Science materials shows significant positive effects on teaching students difficult and dynamic CS concepts [16]. OpenDSA [28][27][18] is an eTextbook system that focuses on visual presentation and interactive exercises [18]. Research has shown positive effects when using the material as course content for Computer Science students in multiple Universities. Using interactive visualization led students to learn CS concepts in a non-passive way. Also, OpenDSA gives students feedback on a graded exercises right away, making interactive learning more achievable. It is not just reading the textbook but also engaging repeatedly with the material, so it raises the efficiency of education and student's interests.

1.2 Problem Statement and Project Goals

CS 1014 Computational Thinking (CT) [22] is an introductory computing course at Virginia Tech aimed at novice learners who have no prior computing experiences. The course covers significant topics in computing such as Algorithms, Abstraction, and Social Impacts. Instructors of the CT course produced a pre-test and three post-tests to investigate and analyze students' knowledge gain of iteration in Spring 2018 [19]. This is elaborated in Section 3.3 and 4.1. Compared to other concepts covered during the semester, students showed a significantly lower acquisition of knowledge for the topic of iteration. In the pre-test and post-tests, we had 12 questions covering various objectives of iteration (see Table 3.1). Individual questions in pre-post tests were systematically edited to be non-identical, but preserved the same inherent conceptual focus (see Appendix A). For example, pre-test Question 3 reads "When using calorie_sum to add up the values in calorie_list, which of the following is the correct initialization for calorie_sum?" Post-test 1 Question 3 reads "When using price_sum to add up the values in price_list, which of the following is the correct initialization for price_sum?"

Research shows that iteration is one of the most difficult topics in cs1 [10]. Dr. Kafura, an instructor of CT course said, "My interactions over several semesters with students in the Computational Thinking class has shown that iteration is a difficult concept for them to learn. Beyond the normal difficulty of mastering programming concepts experienced by novice learners, these students are also generally less quantitatively oriented" [20]. Eric Fouh et. al [18] found that students used interactive visualizations and exercises available in eTextbooks spontaneously before exams or homework. [20] There were many researchers developing iteration visualization to improve iteration materials (see Chapter 2). We hypothesized that interactive visualizations and exercises will improve knowledge gain for non-CS majors on this topic. Therefore, we decided to use interactive visualizations as a part of the CT course.

1.3 Project Management

Jieun Chon, a Master's student, is the main developer of the interactive visualizations and Exercises. Jieun worked with Dr. Shaffer, Dr. Kafura, Dr. Bart, Doctoral student Luke and Doctoral student Javier. Dr. Shaffer is Professor at Virginia Tech, Department of Computer Science in Blacksburg Virginia, who is a supervisor of Jieun Chon and the author of OpenDSA [27]. Dr. Kafura is also co-advisor of Jieun and he is also a Professor of Virginia Tech's Computer Science department. He designed and taught the CT course [22] for two years, including Spring 2018, Fall 2018 and Spring 2019. Dr. Bart is an Assistant Professor at the University of Delaware. He is the developer of the BlockPy [7] and he was at Virginia Tech as an visiting Professor in Spring 2018. He also taught the CT course with Dr. Kafura. Doctoral student Luke Gusukuma was involved in developing automatic feedback on programming problems. Doctoral student Javier Tibau is a new instructor of the CT course in Spring 2019.

For the design of the interactive visualizations, our goal was to show animations of the behavior of BlockPy in developing automatic feedback on programming problems. I developed four different visualizations and two interactive exercises. One exercise shows variable status in the middle of the iteration, which was a big challenge for the students in Spring 2018. The exercise was created by using JavaScript and the JavaScript Algorithm Visualization Library (JSAV) within the Khan Academy Framework [17][3][2][5]. I also developed a second exercise, where goal is to make sure students understand each step in the entire iteration cycle. We will describe the design of the visualizations and exercises in Chapter 3. A variety of means were used to assess the impact of the interactive visualizations on student learning including the pre-post tests, OpenDSA and Canvas usage data, and survey results. This analysis process was conducted to determine the effectiveness of our iteration materials.

1.4 Use of Interactive Visualization

We conducted evaluation with the goal of determining the benefits of using interactive visualizations. We conducted pre-test and three post-tests, and analysed the results to compare the performance in Spring 2018, Fall 2018, and Spring 2019. We tested the hypothesis that there would be increased knowledge gain after updating the course material with interactive visualizations and exercises. We also tried to conduct interviews and surveys with students of the CT classes to collect data and feedback to see if students liked using the interactive visualization. We collected interactive exercise and Canvas page usage data to analyze the students' usage and behavior. We wanted to see how students' interactive usage affects their knowledge gain. In doing this, we also want to help improve the usability of the future development process for interactive visualizations.

The purpose of our study is to encourage students to learn the iteration concept efficiently. This was the first time that OpenDSA [27] was used in an introductory class for novice learners. Also, this research will impact Computer Science education to suggest visualization as a way to teach not only for the CT class at Virginia Tech, but also any CS classes in other universities. Finally, the designs and ideas of these interactive visualizations and exercises are themselves novel, being the first to target block-based languages. This can help further development of other course materials in the future.

Chapter 2

Review of Literature

Iteration is one of the most challenging concepts for novice Computer Science learners. Accordingly, many researchers have implemented various visualization-based course materials. Zhang et. al [30] presented a game-like instructional module titled "Iterative Dungeon" that was used and evaluated in CSC 1310 at Winston-Salem State University (WSSU). They conducted and evaluated pre-test, post-test, survey and interviews. The module visualized and reinforced the *for* loop, *nested for* loop, and *while* loop. However, 17 out of 21 students were CS/IT majors, and the remaining four students were Mathematics, Studio Art, Nursing and Accounting majors respectively. The game was not designed targeting non-CS majors. Students who know typical programming languages such as *Java* or *C++* were the main target of this research. Also, only 21 students took the pre-test and only 17 students took the post tests, which is a small sample size. Moreover, according to the pre-test and post-test comparisons, the researchers did not see the significant learning gains they expected.

Olsson et. al [26] presented visualization as a way to improve learner's control and understanding of programming concepts. They designed and developed a visualization showing program code and the status of variables in a *for*-loop. However, they only conducted surveys of students by asking if the lecture was a clear and effective way of transferring knowledge. So we cannot say that students' knowledge gain was statistically significant. Also, the design targeted students who were learning a text-based programming language, not a block-based language.

Eagle et. al [13][14] of North Carolina State University developed "Wu's Castle", which is a 2-dimensional role-playing game. In this game, the user can move a character by modifying nested loops. The majority of students who used this game during the class gave a positive opinion when surveyed. Also, they showed statistically significant learning gains compared to students who did not play the game. This result encourages developers to consider using visualization technologies in CS course materials. However, there were only 27 students in the class where 16 students played the game and 11 students did not.

Dougherty et. al [12] used Alice [11][6] for CS 0 at Haverford College, where students mainly had no prior programming experiences. Alice is a block-based programming environment that makes it easy to create animations, build interactive narratives, or program simple games. They designed and implemented virtual worlds in Alice to help students learn the iteration concept more easily. They created a virtual world where a boat is looping around on the water. They collected student comments, 13 out of 34 students responded. They also conducted a comparison between two exams, but concluded that there were too many uncontrolled variables to draw any firm conclusions. The research seems a promising approach and showed that visualization can motivate students' interest in the programming language, but the research did not show any statistically significant knowledge gain.

The JavaScript Algorithm Visualization Library (JSAV) is a new JavaScript framework for creating interactive algorithm visualizations [24][25]. JSAV supports not only algorithm visualizations to show how algorithms work step by step, but also supports creating proficiency exercises that students can use to simulate the working of algorithms. Students are given feedback if they make a mistake. The exercise can provide different levels of feedback, so the design can decide the amount of aid to provide students.

BlockPy [7][1] is a web-based Python environment that lets users write programs with blocks or text. BlockPy is designed to support both informal learners and formal class settings. All of its code is entirely open source, and it supports introductory learners as they

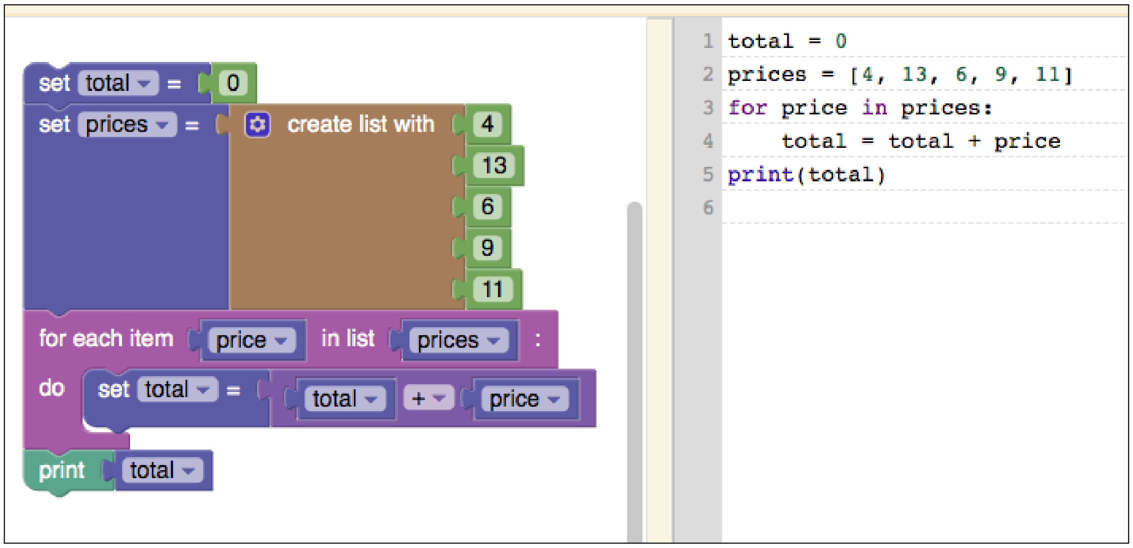

Figure 2.1: Iteration on BlockPy

grapple with Python syntax. Currently, the CT class is using BlockPy for programming assignments. This includes for teaching iteration concepts. Unfortunately, students showed lower knowledge gain on this topic than expectated. BlockPy cannot express some features such as which line is the next step or what is the current status within the iteration process. We expect that our new interactive visualizations of iteration will help CT students, because the new design will show more detail throughout each iteration step.

OpenDSA (an open source, community-based effort to create a complete active eBook for Data Structures and Algorithms) is an integration of text and images with interactive visualizations, simulations, and exercises [27]. JSAV supports many requirements for the development of OpenDSA such as user interface, server-side support, flexibility, visualization control, explanations, engagement, and pseudocode. OpenDSA modules present most of the data structures and core CS concepts such as Big-O analysis, trees, lists, arrays, sorting, file processing, buffer pools, dynamic programming, string algorithms, and others. However, OpenDSA and JSAV did not include content targeted toward teaching iteration in Spring 2018. Thus, we developed interactive visualizations and exercises for iteration.

Chapter 3

Design

There were three design goals: be visually compatible with BlockPy, represent list-based iteration, and develop a scaffolded set of visualizations. We created four visualizations to present different aspects of iteration. Also, two interactive exercises were developed which let students assess their learning of iteration concepts.

All of the visualizations can be used externally. For example, we can generate URLs for each visualization and exercises. The CT course is currently using Canvas, learning management system. Most of the assignments, reading materials, and quizzes for the CT class are presented as course pages at the Canvas site. By creating a URL for each, instructors were able to put each visualization in the right place within the course content. In this way, students did not need to go to an external link to access the material. This accessibility allowed for user-friendly access, and most of the students actually used the visualizations and exercises. Figure 3.1 shows one of the visualizations embedded in a reading assignment on the Canvas page. Figure 3.6 and 3.7 show exercises on the Canvas page that were assigned on Day 8 of the course.

3.1 Interactive Visualizations

Figure 3.2 shows three main aspects of the interactive visualizations: navigators, descriptions, and parts. The animation parts include the representations of the BlockPy coding

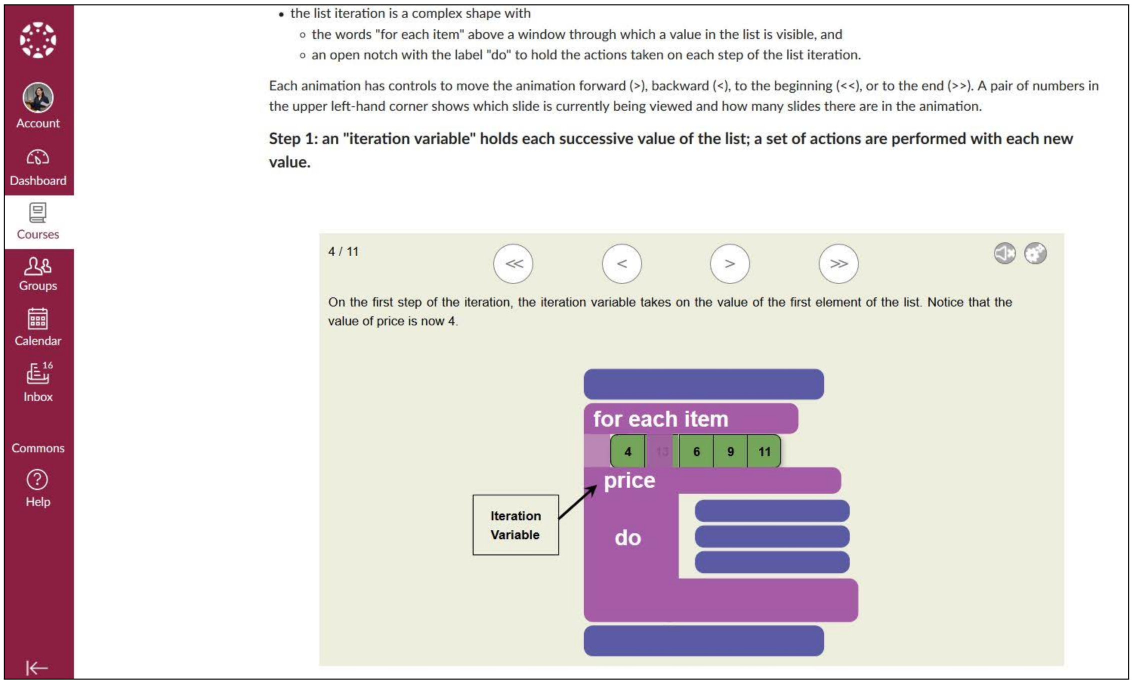

Figure 3.1: A Canvas reading assignment with embaded extend visualization

blocks, variables and their values, and the console output area. By clicking the navigation buttons, students can go forward, back, to the beginning, and to the end of the visualization. The description is up to three lines for each slide, which gives the essential explanation for each step. The blocks can move, highlight, appear or disappear, depending on each step's behavior, so that students can see exactly what is happening.

All of our the visualizations are designed to be similar in appearance to the blocks in BlockPy [8][7], which students are currently using as their main programming language for the CT. We tried to match the shape, color and structure to make our visualizations look same as much like BlockPy as possible. In this way, when the students encounter BlockPy problems, they can recall the images of the visualizations and the exercises in their mind, which will assist them in arriving at the correct answers.

The first visualization (see Figure 3.2) illustrates the basic execution sequence of iteration. It shows how a *for* loop works in broad outline. It avoids details, focusing on the concept.

We expect that students can draw an initial image of the iteration concept in their mind after using the first visualization. Then, we added more details of the iteration process to the second visualization.

The second visualization (see Figure 3.3) focused on the status of the value of the iteration

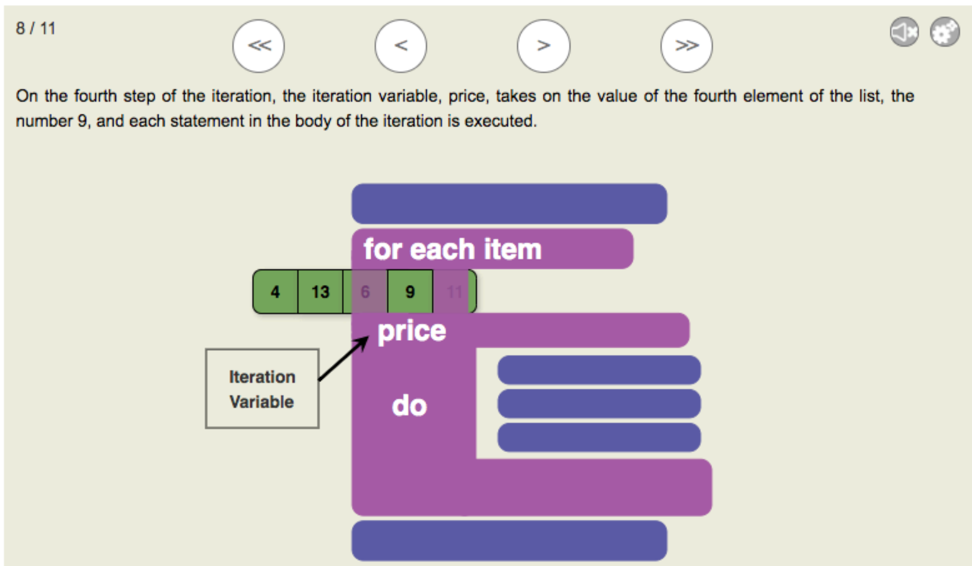

Figure 3.2: First Interactive Visualization.

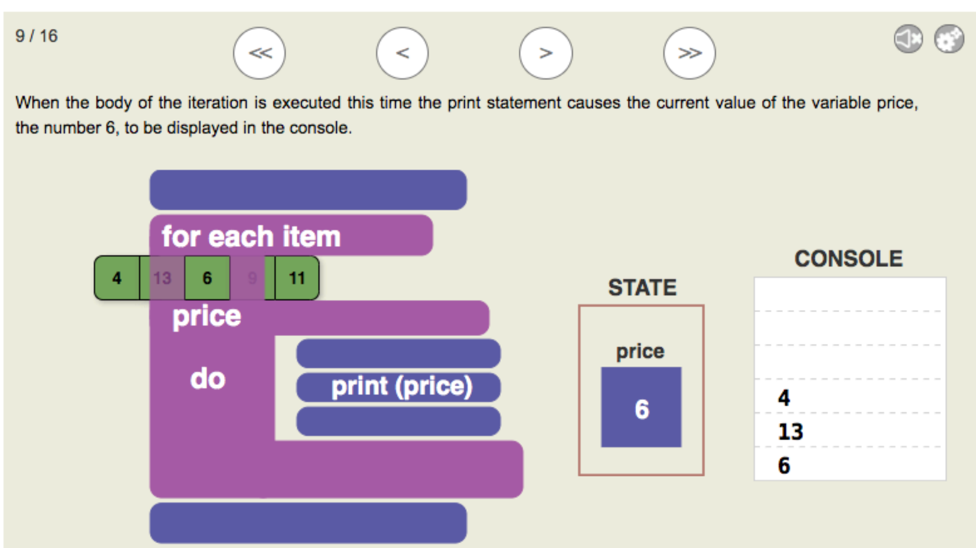

Figure 3.3: Second Interactive Visualization.

variable that we are using within the iteration body. We created a box next to the program blocks that shows status changes at every step. We also added a console box next to the status box. This mimics BlockPy's console box design so that students will assume the visualization represents BlockPy. For each cycle of the *for* loop, we printed out the current element value in the array (the price), so that students will understand how the value of the iteration variable is changed.

The third visualization (see Figure 3.4) adds calculation within the iteration. It focuses on

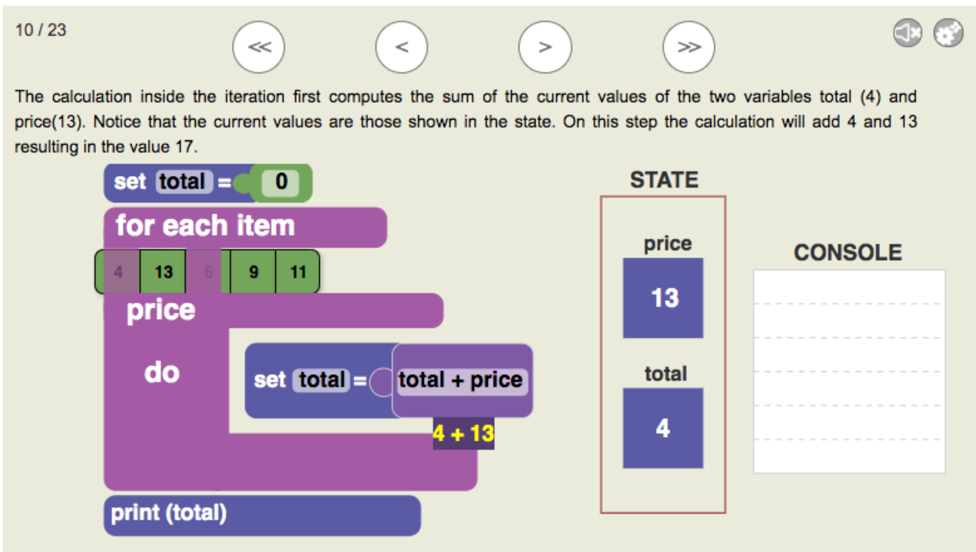

Figure 3.4: Thrid Interactive Visualization.

when exactly the value is changed, so we showed students how to calculate the sum of all of the list values within the iteration.

For the last visualization, we created a more complicated calculation (see Figure 3.5), that will give students a challenge to keep track of more variables than previous visualizations. They will learn when each value's status for each step changes. This visualization calculates the average of a list, showing the entire process of calculating total and count values, and how these values increase at each stage.

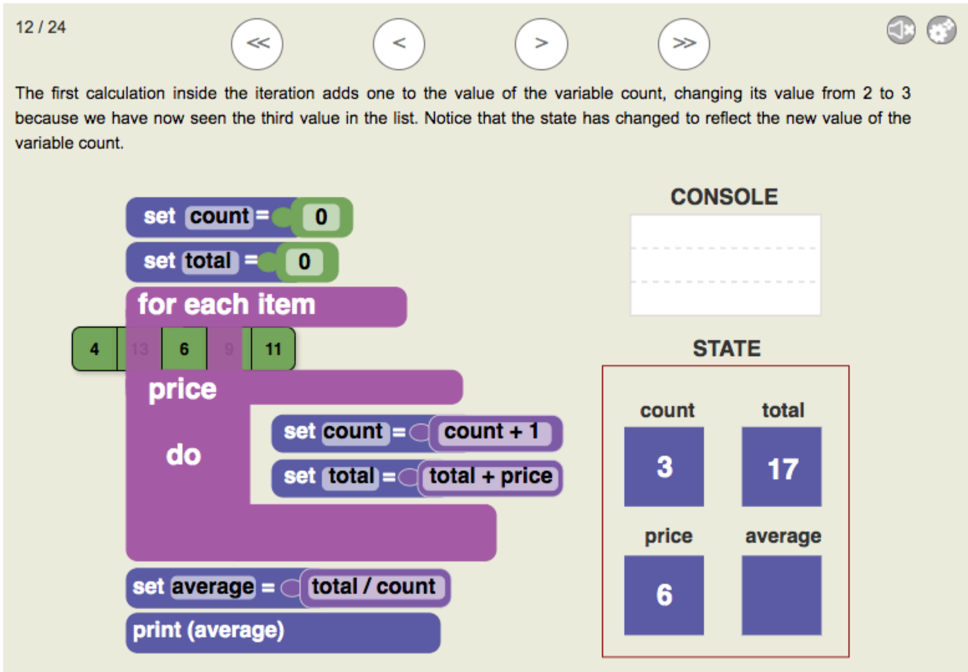

Figure 3.5: Fourth Interactive Visualization.

3.2 Exercises

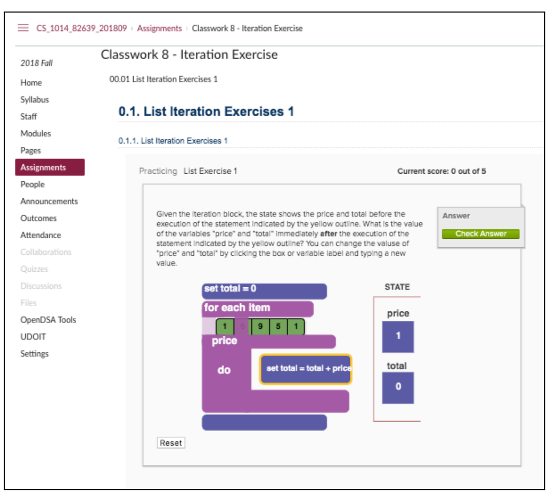

Figure 3.6: A Khan Exercise in Canvas.

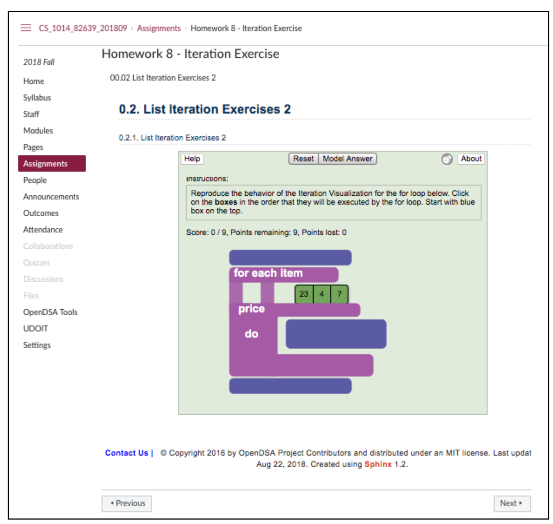

Figure 3.7: A JSAV Exercise in Canvas.

We made two types of exercises to give students feedback on their mastery of iteration concepts. We used the Khan Academy Exercise Infrastructure [5][2] to create an exercise covering the state of values during the iteration. We will refer to this as the Khan Academy Exercise. For the design of the Khan Academy Exercise, we mimicked BlockPy's design. Figure 3.6 shows the Khan Exercise that we developed to test students on the materials shown in Figure 2.1. The exercise was designed to ask students about status of values in the middle of the iteration body. Specifically, we asked about the status change after one execution in the iteration body. This concept was one of the biggest challenges for the students in Spring 2018. We made one type of question but with randomized list size and values, so that students can get more opportunities for practice.

For example, as Figure 3.6 shows, the question is asking the status of price and total when the index is randomly chosen at position, 1. The size of the list was random, between 3 to 7, and the current position is random. The variables in the list were also generated randomly from 1 to 9. If the student clicks the box or the number inside of the state area, they can change the values by typing the new price and total (see Figure 3.8). Each student was given a randomized problem instance and we graded the results using our automated grading system. On Day 8, students were encouraged to complete the Khan Academy Exercise during the class, but they could submit the assignment anytime before Day 10 (see Figure 3.15). Students had to answer five exercises correctly, and every time they gave correct answers or clicked the reset button, we generated a new problem instance. The students were granted an unlimited amount of chances to complete the exercise. If the students gave five correct answers, they received full credit for the assignment. Also, if they made wrong answers, we did not subtract from the total score, but we gave them feedback saying the answer was wrong.

We also created another exercise using the JSAV Exercise Framework [25][24][23], that we will refer to as the JSAV exercise. The structure is similar to the first interactive visualization (see Figure 3.2). The question focused on understanding the order of execution in an

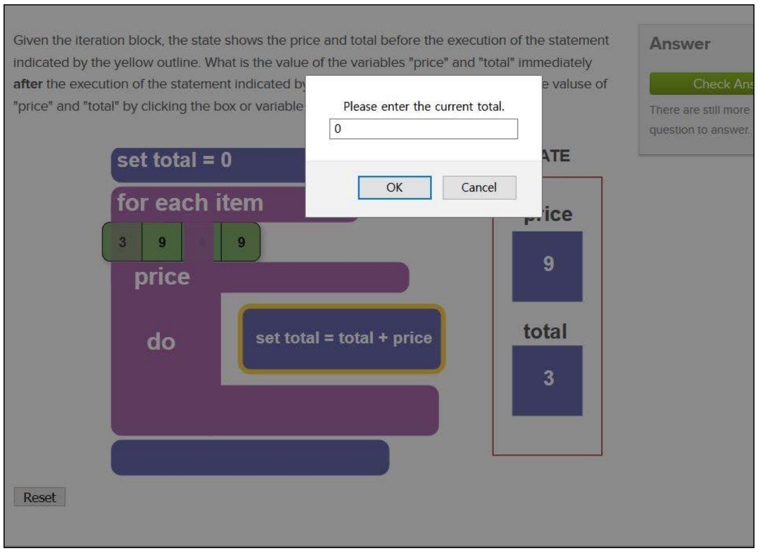

Figure 3.8: Khan Exercise Entering Value

iteration. In this exercise, we expected students to understand the four parts of the iteration: initialization (see Figure 3.11), iteration statement (see Figure 3.12), iteration body (see Figure 3.13) and execution after the iteration (see Figure 3.14). If they hover the mouse over each part, it shows a yellow bold line, so that they know which section they are about to click. Also, students were expected to know the order of executions within the iteration body. Students were asked to click one box at each time in the entire iteration cycle. Like the Khan Academy exercise, we generated a list size (3 to 5) with random values inside (1 to 24). We also made a Model Answer, as Figure 3.9 shows. However, if the student views the Model Answer, they can no longer receive credit for this instance of the exercise. To receive full credit, students had to reset or refresh the page to get a new problem instance, and complete that correctly. On Day 8, students were recommended to complete the JSAV Exercise during class time, but they could also submit the assignment before Day 10 (see Figure 3.15). To get full credit, students could not miss any of the steps. If they miss any, we gave them feedback that they missed a step, and we took a point off from the exercise score (see Figure 3.10). If the students successfully completed the exercise, they received full credit for the assignment. That means if a student missed any of the steps, they could choose to re-do the assignment with a new generated list to try for full credit. Students were allowed to take the JSAV Exercise as many tries as they wanted.

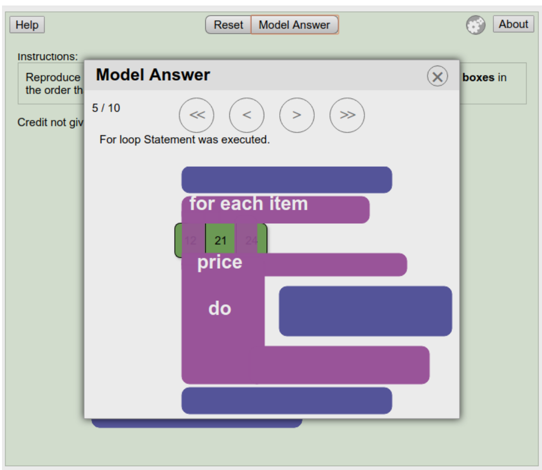

Figure 3.9: JSAV Exercise Model Answer

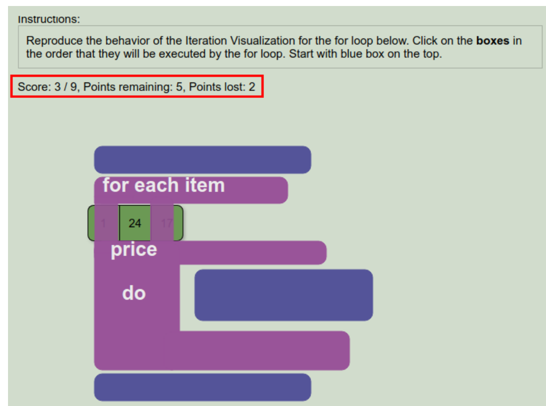

Figure 3.10: JSAV Exercise Grading

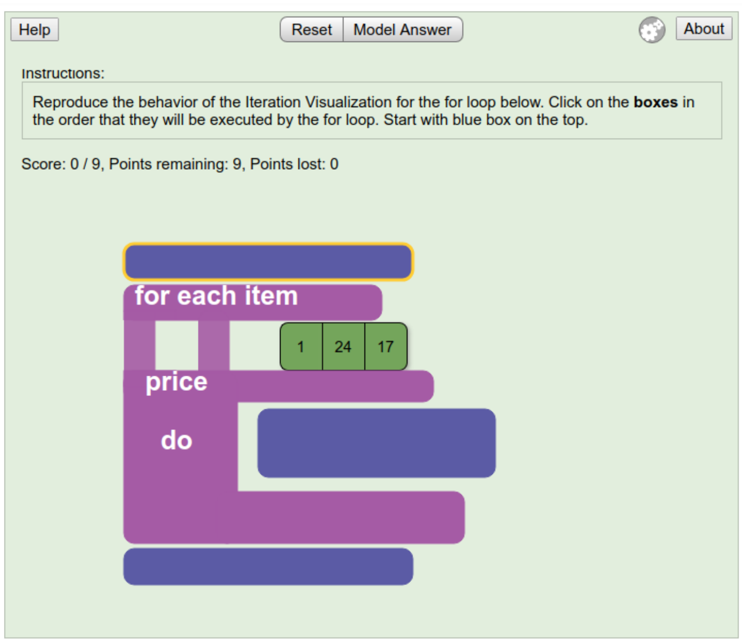

Figure 3.11: JSAV Exercise, hovering over the initialization part. Notice the yellow outline for the top block in the image.

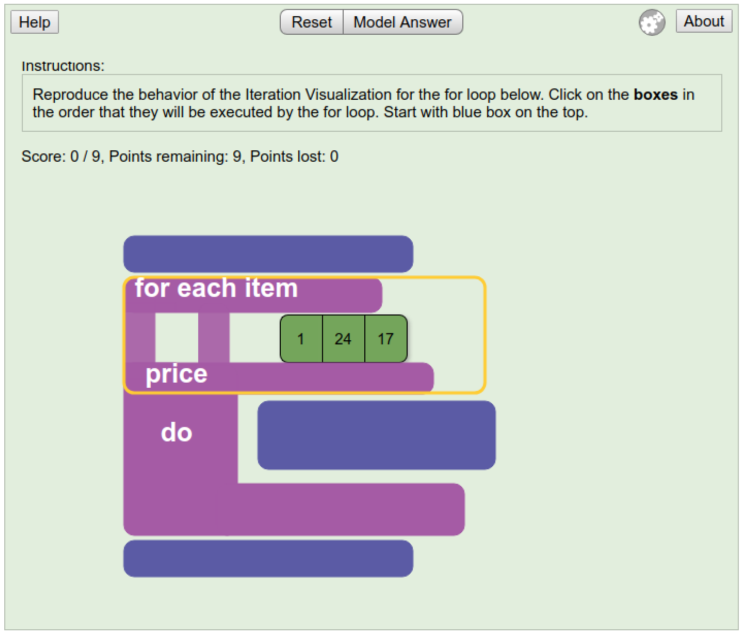

Figure 3.12: JSAV Exercise, hovering over the iteration statement part.

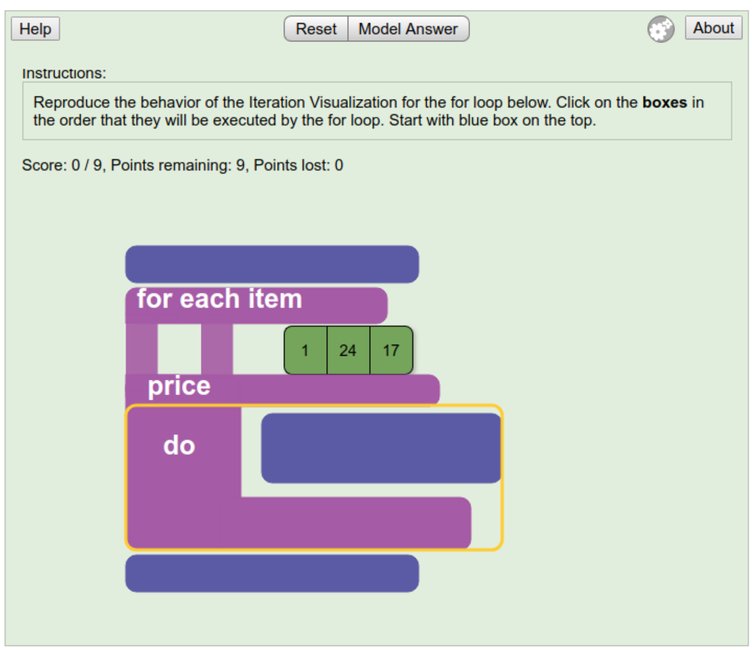

Figure 3.13: JSAV Exercise hovering over the iteration body part.

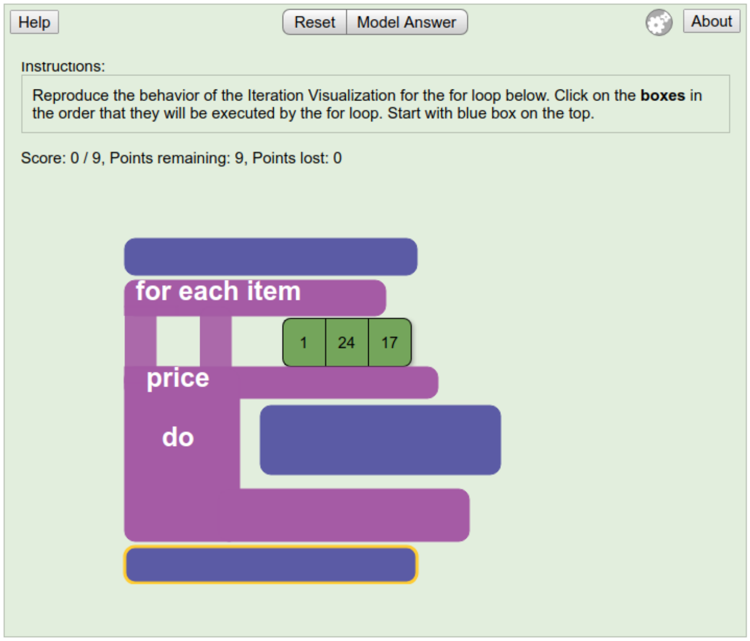

Figure 3.14: JSAV Exercise Hovering over the part outside of the iteration. Notice the yellow outline for the top block in the image

3.3 Pre-Test and Post-Test Design

For each semester, the course used identical readings, quizzes, classwork and homework problems, and exercises. The pre-test and three post-tests were designed by Dr. Kafura, Dr. Bart, and Doctoral student Luke Gusukuma. The design process was part of Luke's research [19]. Each semester, we conducted a pre-test during Day 7 of the class at the beginning of

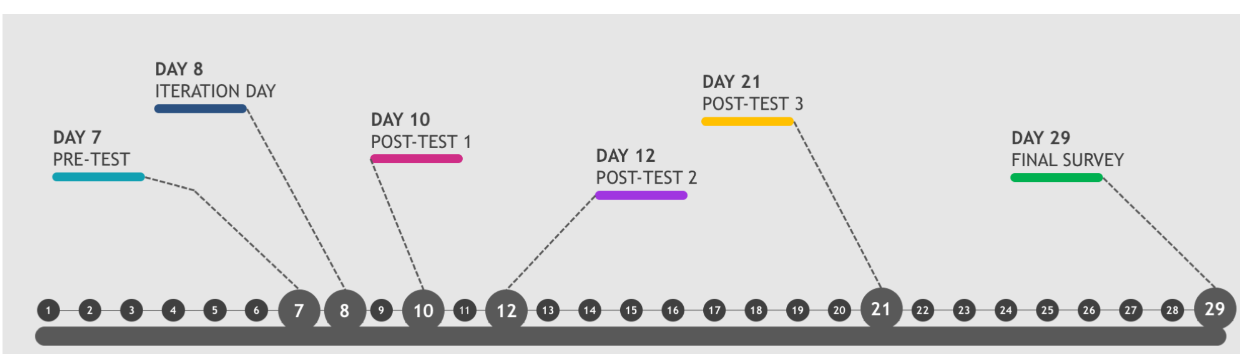

Figure 3.15: Planned Timeline

	Objective
Q1	Type of element within the list
Q2	Correctly place a block in the iteration
Q3	How to initialize variable to use in the iteration
Q4	Sum in the body of the iteration
Q5	Relationship between the iteration variable and the lists' element
Q6	Correct structure to include the sum in the iteration
Q7	Initialization of the empty list variable
Q8	Correct algorithm to form a new list of elements that satisfy some criteria
Q9	Create the new list by modifying the elements in existing list
Q10	Concept of state during the iteration
Q11	Concept of state during the iteration
Q12	Scope of the iteration

Table 3.1: CT Course Pre-Post Test Objective

the class for about 20 minutes (see Figure 3.15). Also, students were asked to take three post-tests after Day 8. Post-test 1 was given in Day 10 of the class, Post-test 2 was given on Day 12, and the Post-test 3 was given on Day 21. We conducted three different post-tests to determine whether iteration knowledge persists.

All pre-post tests are the same for all three semesters, to make the condition equal for all groups. The post-test questions held the same content as the questions on the pre-test. For instance, on the pre-test one value may be price_list but on the post-test, it might read as weight_list. The full set of pre-post test questions can be found in Appendix A. The first three quizzes, pre-test, post-test 1, and post-test 2, asked using BlockPy and post-test 3 asked questions about equivalent Python syntax.

3.4 OpenDSA Data Analysis

We conducted usage analysis as our second evaluation method. When students used interactive visualizations and exercises, the system automatically collected user behavior data. The data contained the start time, end time, score, and other timing data. Also, we were able to see the amount of time students tried exercises or made incorrect answers. By using this method, we tried to answer the following research questions:

- Will students use interactive visualizations and exercises more than one time? What will be the average usage amount?
- Will students use interactive visualizations and exercises to help study for the quizzes?
- What will be the average length of usage for each visualization and exercise? What is the distribution of usage among students?
- Will students who use interactive visualizations and exercises perform better than the students who do not?
- Will students who used the interactive visualizations perform better on questions related to the dynamic execution of iteration (quiz question 10-12)?
- What will be the average number of attempts to complete the Khan exercise and JSAV exercise?
- How many students pressed the previous button on the visualization to go back to the previous status of the animation?
- How many students will skip the visualizations, and what is their performance?

3.5 Canvas Data Analysis

We also conducted another data analysis using Canvas usage data. We collected data of the iteration reading assignment page that contains our visualizations to see students' usage behavior in Fall 2018 and Spring 2019. We tried to answer the following research questions:

- Will students use the reading assignment page more than one time?
- What will be the average usage amount?
- Will students who use the reading assignment page more than once show a better grade than the students who do not?

Chapter 4

Evaluation

This section presents the evaluation for the interactive visualizations of iteration. For each semester, we will explain the research timeline, class demographics, major demographics, participants, and results from the pre-post tests. For the T-Test analysis section, we examined the knowledge gains for all questions and for Question 10 - 12 (see Table 3.1 and Appendix A). We discuss performance on quizzes, and what the test results indicate regrading which questions students found the most difficult or easy. We also cover our findings from analysing OpenDSA and Canvas data, and from the student surveys.

4.1 Spring 2018

4.1.1 CT Course in Spring 2018

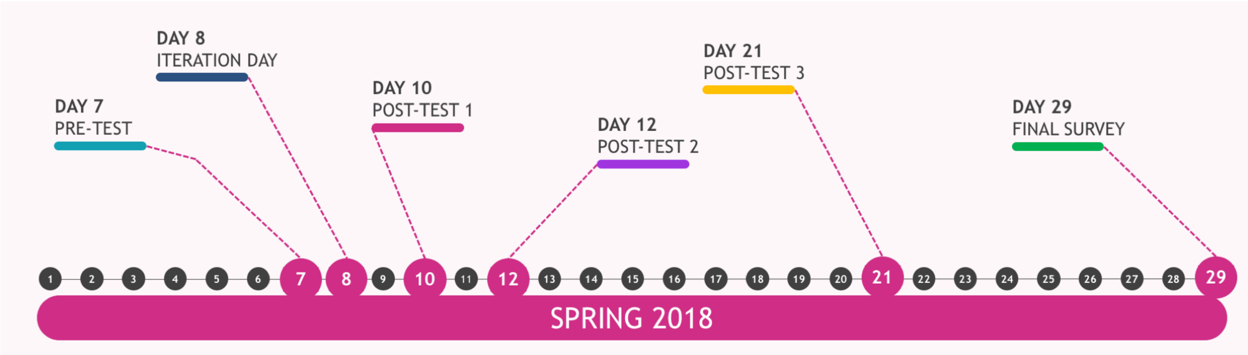

Figure 4.1: CT Course Timeline on Spring 2018

In Spring 2018, Drs. Kafura and Bart were the instructors of the CT courses. Dr. Kafura's class had 75 students and Dr. Bart's class had 62 students. Both classes were taught with the same material and all the course designs were the same, but the class time was different. Also, both classes conducted a pre-post test before and after Day 8, covering iteration concepts, to see if students were showing expected knowledge gains for the subject.

Figure 4.1 shows the timeline for Spring 2018. We conducted the pre-test on Day 7, covered list-based iteration on Day 8, conducted Post-test 1 on Day 10, conducted Post-test 2 on Day 12, conducted Post-test 3 on Day 21, and conducted a survey on Day 29. This research was carried out under IRB #16-502, Improved Learning of Introductory Computing Concepts. We received consent to analyze data from 126 of 137 students, or 92% of the class. As Table 4.1 shows, Spring 2018 semester included students from a variety of majors as summarized in Tables 4.1 and 4.2. Only 10% of the students were in Science, Technology, Engineering and Math (STEM) related majors. We had 1 Biological Sciences student, 1 Chemistry student, 1 Clinical Neuroscience student, 1 Computational Modeling and Data Analytics student, 2 Computational and Systems Neuroscience students, 1 Explore Technology student, 3 General Engineering students, 2 Mathematics students, and 1 Systems Biology student, . About 39% of the students were Sophomores, 24% were Juniors, 19% were Freshman, and the remaining 18% were Seniors. There were also 8 honors students (6%).

Table 4.2: Spring 2018 Major Demographics

Major	**Count**	%	STEM
Art (Fine Arts)	2	2%	
Biological Sciences	1	1%	YES
Building Construction	25	20%	
Business, Undecided	2	2%	
Chemistry	1	1%	YES

Table 4.2 – continued from previous page

Major	Count	%	STEM
Cinema	1	1%	
Clinical Neuroscience	1	1%	YES
Communication Studies	1	1%	
Comp Model & Data Anyl	1	1%	YES
Computational & Sys Neursci	2	2%	YES
Creative Writing	1	1%	
Criminology	14	11%	
Dairy Science	3	2%	
Economics, Business	1	1%	
Explore Technology	1	1%	YES
Fashion Merch. and Design	6	5%	
General Engineering	3	2%	YES
Geography	2	2%	
History	2	2%	
Human Development	1	1%	
International Relations	1	1%	
International Studies	2	2%	
Literature and Language	1	1%	
Mathematics	2	2%	YES
Meteorology	2	2%	
Multimedia Journalism	5	4%	
Music	1	1%	
Nat's Security & Foreign Aff	2	2%	

Table 4.2 – continued from previous page

Major	Count	%	STEM
Packaging Systems & Design	1	1%	
Political Science	6	5%	
Prof & Tech Writing	2	2%	
Property Management	1	1%	
Psychology	9	7%	
Public and Urban Affairs	1	1%	
Public Relations	2	2%	
Sociology	1	1%	
Systems Biology	1	1%	YES
Theatre Arts	1	1%	
University Studies	12	10%	
Total	124		9

4.1.2 Analysis

Three paired-samples t-tests were conducted to compare the knowledge level of iteration before and after learning iteration material on Day 8 for the Spring 2018 studies (see Table 4.3). There were total 137 students in the class and only 126 students signed the consent form. There were 119 students participated in the pre-test, but only 112 of those students signed the consent form and 18 students did not participate. For post-test 1 and post-test 2, 119 students participated, but 9 of those students did not sign the consent form and

		Section 1 (Kafura)		Section 2 (Bart)		Total	
		Count	%	Count	%		%
Major	STEM	6	9%	9	16%	13	10%
	Other	61	91%	48	84%	111	90%
Year	Freshman	11	16%	13	23%	24	19%
	Sophomore	26	39%	22	39%	48	39%
	Junior	17	25%	13	23%	30	24%
	Senior	13	19%	9	16%	22	18%
Honor Students		4	6%	4	7%	8	6%
Number of Students		67		57		124	

Table 4.1: Spring 2018 Class Demographics

	Signed Consent Form	Participants
Pre-test	112	119
Post-test 1	110	119
Post-test 2	110	119
Post-test 3	99	105
Total	126	137

Table 4.3: Spring 2018 Participants Demographics

18 students did not participate. This does not mean that the 18 students who did not participate were the same for each test. For post-test 3, 105 students participated, only 99 of those students signed the consent form, and 22 students did not participate. As stated earlier, these students may not be the same as the previous 18 students for the other post-tests. Absence from Day 7, Day 10, Day 12, and Day 21 was the main reason why students chose not to do the pre-test or post-tests.

4.1.3 Pre-Post Test Result

The following is a summary of the most relevant findings of the experiment. When we say that a result is significant, we mean that the p-value is less than 0.05. We conducted the pre-test on Day 7, Post-test 1 on Day 10, Post-test 2 on Day 12, and Post-test 3 on Day 21 (see Figure 4.1). Statistically, there was a significant difference in the scores for the pre-test (M=7.03, SD=2.49) and Post-test 1 (M=9.01, SD=2.49); t(101)=11.41, p =7.13E-20. Also, there was a significant difference in the scores for the pre-test (M=6.93, SD=2.56) and Post-test 2 (M=9.33, SD=1.97); t(101)=13.56, p =1.74E-24. Finally, there was a significant difference in the scores for the pre-test (M=6.96, SD=2.47) and Post-test 3 (M=9.28, SD=1.91); t(92)=11.67, p =7.46E-20. Table 4.4, 4.5, and 4.6 shows the full set of T-Test results for Spring 2018. However, the knowledge gain was lower than other topics in the CT course. This motivated us develop an interactive visualization for future semesters.

As Table 4.7 shows, students missed Question 11 the most. This question tests student on understanding of the concept of state during the iteration. Only 27% of students were correct on the pre-test, 48% were correct on Post-test 1, 33% were correct on Post-test 2, and 30% were correct on Post-test 3. Students showed the most knowledge gain on Question 4, which asked about the summation process in the body of the iteration (see Table 3.1).

We also conducted a t-test only for Questions 10 - 12. The following is a summary of the most relevant findings of the experiment. When we say that a result is significant, we mean that the p-value is less than 0.05. Statistically, there was a significant difference in the scores of Question 10-12 for the pre-test (M=1.37, SD=0.90) and Post-test 1 (M=1.72, SD=0.89); t(101)=3.92, p =1.64E-04. Also, There was a significant difference in the scores for the pre-test (M=1.34, SD=0.90) and Post-test 2 (M=1.61, SD=0.84); t(101)=2.93, p =4.24E-03. There was a significant difference in the scores for the pre-test (M=1.37, SD=0.89) and the Post-test 3 (M=1.66, SD=0.82); t(92)=3.08, p =2.77E-03. Tables 4.8, 4.9, and 4.10 show full T-Test results for the Questions 10-12 in Spring 2018. Even though the improvement was significant, the correctness was much lower than other questions.

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	post-test1	9.01	102	2.11	0.21
	pre-test	7.03	102	2.49	0.25
Pair 2	post-test2	9.33	102	1.97	0.20
	pre-test	6.93	102	2.56	0.25
Pair 3	post-tet3	9.28	93	1.91	0.20
	pre-test	6.96	93	2.47	0.26

Table 4.4: Spring 2018 Paired Samples Statistics

Paired Samples Correlations				
		N	Correlation	Sig.
Pair 1	post-test1 & pre-test	102	0.72	8.84E-18
Pair 2	post-test2 & pre-test	102	0.72	2.08E-17
Pair 3	post-tet3 & pre-test	93	0.64	3.87E-12

Table 4.5: Spring 2018 Paired Samples Correlations

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	post-test1 - pre-test	1.98	1.75	0.17	1.63	2.32	11.41	101	7.129E-20
Pair 2	post-test2 - pre-test	2.40	1.79	0.18	2.05	2.75	13.56	101	1.737E-24
Pair 3	post-tet3 - pre-test	2.33	1.92	0.20	1.93	2.72	11.67	92	7.459E-20

Table 4.6: Spring 2018 Paired Samples Test

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
pre-test	64%	87%	58%	31%	89%	70%	51%	40%	64%	46%	27%	60%
post-test 1	59%	97%	94%	78%	98%	84%	68%	65%	79%	62%	48%	59%
post-test 2	71%	96%	91%	74%	96%	87%	85%	82%	85%	65%	33%	60%
post-test 3	71%	98%	94%	72%	97%	82%	86%	68%	83%	73%	30%	64%

Table 4.7: Spring 2018 Test Correctness

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	post-test1	1.72	102	0.89	0.09
	pre-test	1.37	102	0.90	0.09
Pair 2	post-test2	1.61	102	0.84	0.08
	pre-test	1.34	102	0.90	0.09
Pair 3	post-tet3	1.66	93	0.82	0.08
	pre-test	1.37	93	0.89	0.09

Table 4.8: Spring 2018 Q10-12 Paired Samples Statistics

Paired Samples Correlations				
		N	Correlation	Sig.
Pair 1	post-test1 & pre-test	102	0.494	1.36E-07
Pair 2	post-test2 & pre-test	102	0.449	2.19E-06
Pair 3	post-tet3 & pre-test	93	0.409	4.63E-05

Table 4.9: Spring 2018 Q10-12 Paired Samples Correlations

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	post-test1 - pre-test	0.35	0.90	0.09	0.17	0.52	3.92	101	1.6.E-04
Pair 2	post-test2 - pre-test	0.26	0.91	0.09	0.09	0.44	2.93	101	4.2.E-03
Pair 3	post-tet3 - pre-test	0.30	0.93	0.10	0.10	0.49	3.08	92	2.8.E-03

Table 4.10: Spring 2018 Q10-12 Paired Samples Test

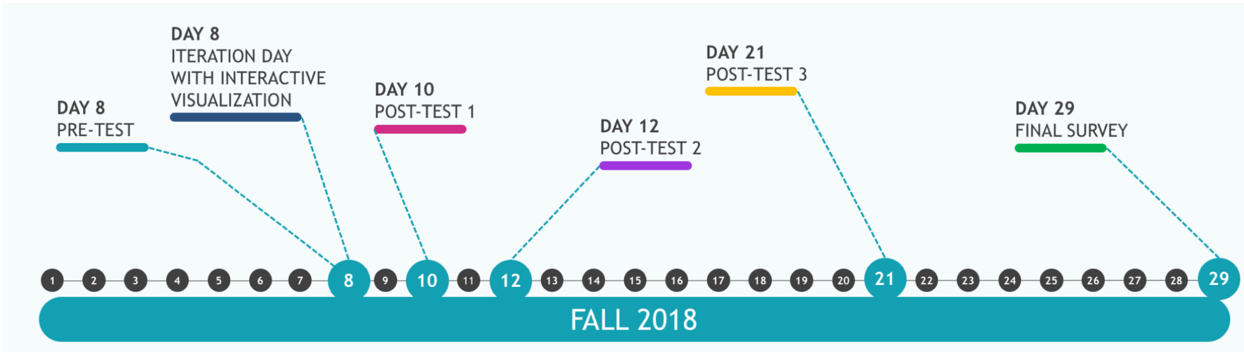

Figure 4.2: CT Course Timeline on Fall 2018

4.2 Fall 2018

4.2.1 CT Course in Fall 2018

In Fall 2018, we used new interactive visualizations to present the topic of iteration. We added three interactive visualizations on the course reading assignment page, which students were recommended to read before attending class. We conducted again a set of pre-post tests before and after Day 8 when this topic is covered in class. The pre-test was given on Day 8, post-test 1 on Day 10, post-test 2 on Day 12, and post-test 3 on Day 21 (see Figure 4.2). Students were asked to take these tests at the beginning of the class. The questions were identical to those given in Spring 2018, so we can compare. The two sections were taught by Dr. Kafura. We collected and analyzed roster data from 120 of the 127 students (94%). Fall semester included students from a variety of majors as summarized in Tables 4.11 and 4.12. One section had a small number of STEM majors similar to Spring 18. The other section had more students who were STEM majors. In Spring 2018, we had only 10% of students who were STEM major, but in Fall 2018, 33% were in STEM majors. We had 1 Business Information Technology students, 4 Computational Modeling and Data Analytics students, 4 Explore Technology students, 11 General Engineering students, 1 Mathematics student, and 15 Statistics students. The students were 40% Freshmen, 28% Sophomores,

		Section 1 (Kafura's)		Section 2 (Kafura's)		Total	
		Count	%	Count	%	Count	%
Major	STEM	12	19%	27	47%	39	33%
	Other	51	81%	30	53%	81	68%
Year	Freshman	17	27%	31	54%	48	40%
	Sophomore	19	30%	14	25%	33	28%
	Junior	18	29%	9	16%	27	23%
	Senior	9	14%	3	5%	12	10%
Honor Students		3	5%	3	5%	6	5%
Number of Students		63		57		120	

Table 4.11: Fall 2018 Class Demographics

23% Juniors, and 10% Seniors. There were also 6 honors students (5% of students) (see Table 4.11).

Table 4.12: Fall 2018 Major Demographics

Major	**Count**	%	STEM
Art (Fine Arts)	6	5%	
Biochemistry	1	1%	YES
Building Construction	17	14%	
Business Information Tech	1	1%	YES
Business, Undecided	2	2%	
Cinema	3	3%	
Communication Studies	2	2%	
Comp Model & Data Anyl	4	3%	YES
Creative Writing	1	1%	
Criminology	7	6%	
Economics, Science	1	1%	
Explore Technology	4	3%	YES

Table 4.12 – continued from previous page

Major	**Count**	**%**	STEM
Forestry	1	1%	
General Engineering	11	9%	YES
Geography	4	3%	
History	1	1%	
Human Development	1	1%	
International Relations	3	3%	
International Studies	1	1%	
Liberal Arts (Undecided)	2	2%	
Literature and Language	4	3%	
Mathematics	1	1%	YES
Multimedia Journalism	1	1%	
Nanoscience	1	1%	YES
Nat's Security & Foreign Aff	2	2%	
Physics	1	1%	YES
Political Science	2	2%	
Psychology	11	9%	
Public Relations	2	2%	
Smart & Sustain Cities	1	1%	
Statistics	15	13%	YES
University Studies	6	5%	
Total	120		36

4.2.2 Participants

	Signed Consent Form	Participants
pre-test	111	118
post-test 1	112	118
post-test 2	108	114
post-test 3	108	115
survey	113	118
Total	120	127

Table 4.13: Fall 2018 Participants Demographics

We collected and analyzed pre-post test data from 120 of 127 students (94% of the class) who signed the consent form (see Table 4.13). Three paired-sample t-tests were conducted to compare the knowledge level of iteration on Day 8 and after learning iteration material on Day 8. A total of 118 students participated in the pre-test and 118 students participated in Post-test 1. This does not mean that the 9 students who did not participate were the same for each test. A total of 114 students participated in Post-test 2 and 115 students participated in Post-test 3.

4.2.3 Pre-Post Test Result

In Fall 2018, we conducted a pre-test on Day 8, Post-test 1 on Day 10, Post-test 2 on Day 12, and Post-test 3 on Day 21 (see Figure 4.2). Students were asked to go to the CT Course's canvas page and complete the tests during class within 20 minutes. We had the same structure and questions for the pre-test and ost-tests as Spring 2018 to make the condition equal. The full set of questions asked can be found in Appendix A. When we say that a result is significant, we mean that the p-value is less than 0.05.

There was a significant difference in the scores between the pre-test (M=7.97, SD=2.32) and Post-test 1 (M=9.88, SD=1.88); t(105)=10.24, p =1.77E-17. Also, there was a signifi-

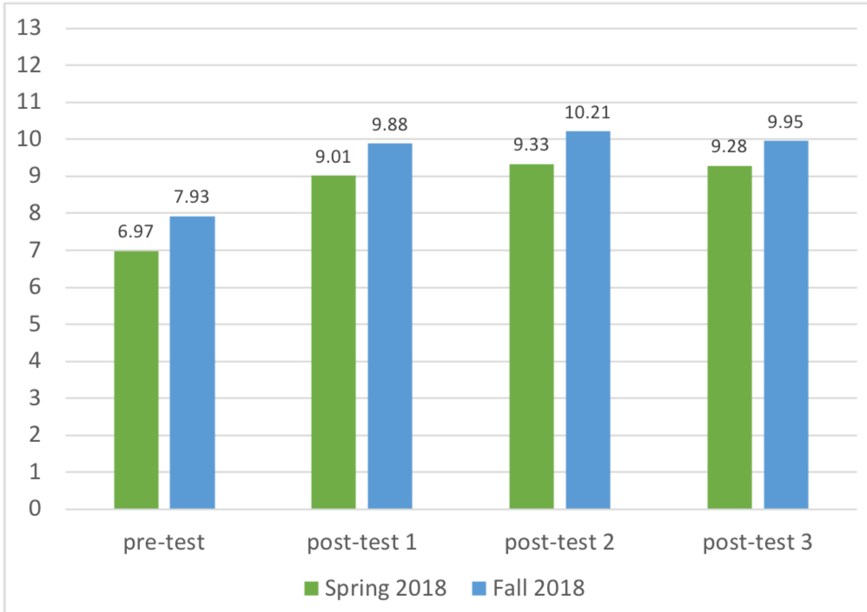

Figure 4.3: Mean Comparison Spring 18 and Fall 18

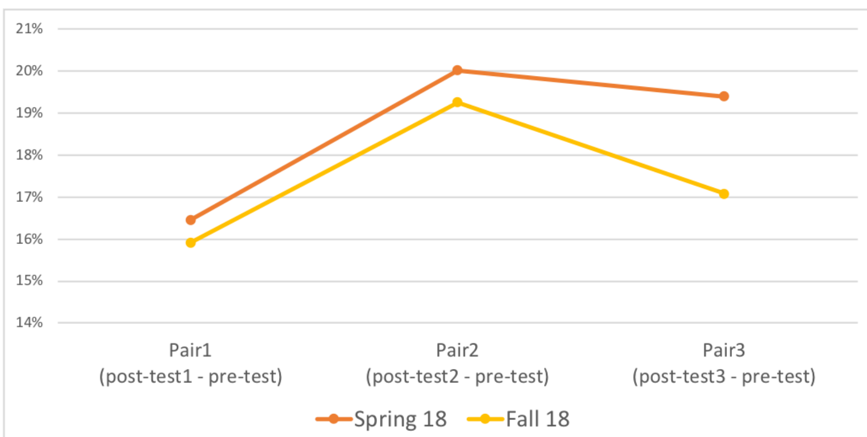

Figure 4.4: Knowledge Gain Comparison Spring 18 and Fall 18

cant difference in the scores between pre-test (M=7.91, SD=2.28) and Post-test 2 (M=1021, SD=1.36); t(102)=13.44, p =2.58E-24. There was a significant difference in the scores for pre-test (M=7.90, SD=2.23) and Post-test 3 (M=9.95, SD=1.70); t(101)=10.46, p =8.67E-18. Tables 4.14, 4.15, and 4.16 show the T-Test details. We found that the means for the post-tests were all much higher than the mean for the pre-test, but the overall result of Fall 2018 showed lower knowledge gain than Spring 2018.

First notice from the test results that the overall score from all post-tests was much higher for Fall 2018 than Spring 2018 (see Figure 4.3). Unfortunately, we cannot say that the conditions were the same between the two-semesters because the Fall 2018 pre-test average was 66.42%, which is much higher than the pre-test score from Spring 2019 which was 57.59%. Consequently, the knowledge gain for Fall 2018 was smaller than for Spring 2018. We believe that this was possibly caused by two factors. First, we conducted the pre-test on Day 8 of the class in Fall 2018, not Day 7. Students were recommended to complete reading assignments before attending class, so it is a likely that the majority of the students were influenced already before they took the pre-test on Day 8. We will elaborate about this in Section 4.3.5. Also, about 30% of the class were STEM majors in Fall 2018, which is higher than Spring 2018, where there were only 10% STEM majors (see Table 4.1 and 4.11). However, we cannot confirm that was the factor. It is dangerous using major as an indicator of better ability because we do not know about their prior experience in computing or programming. On the other hand, we found that students showed higher knowledge gain for Question 10 - 11 in Fall 2018 than Spring 2018. Questions 10 - 11 cover a concept that our exercises were designed to target. As Figure 4.6 shows, Pair 1, between pre-test and post-test 1, of Fall 2018 (M = 0.42, SD = 0.88) showed greater knowledge gain than Spring 2018 (M = 0.35, SD = 0.26), but the effect size was small as d = 0.0786. Pair 2, between pre-test and post-test 2, of Fall 2018 (M = 0.36, SD = 0.91) showed greater knowledge gain than Spring 2018 (M = 0.26, SD = 0.91), but the effect size was small as d = 0.1098. Pair 3, between pre-test and post-test 3, of Fall 2018 did not showed greater knowledge gain than Spring 2018. This result shows that students showed greater knowledge gain right after Day 8 but showed lower knowledge gain at the time of post-test 3.

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	post-test 1	9.88	106	1.88	0.18
	pre-test	7.97	106	2.32	0.23
Pair 2	post-test 2	10.21	103	1.36	0.13
	pre-test	7.91	103	2.28	0.22
Pair 3	post-test 3	9.95	102	1.70	0.17
	pre-test	7.90	102	2.23	0.22

Table 4.14: Fall 2018 Paired Samples Statistics

Paired Samples Correlations				
		N	Correlation	Sig.
Pair 1	post-test 1 & pre-test	106	0.60	1.23E-11
Pair 2	post-test 2 & pre-test	103	0.65	1.75E-13
Pair 3	post-test 3 & pre-test	102	0.52	1.75E-08

Table 4.15: Fall 2018 Paired Samples Correlations

Paired Samples Test									
Pairs		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	post-test 1 - pre-test	1.91	1.92	0.19	1.54	2.28	10.24	105	1.77E-17
Pair 2	post-test 2 - pre-test	2.31	1.74	0.17	1.97	2.65	13.44	102	2.58E-24
Pair 3	post-test 3 - pre-test	2.05	1.98	0.20	1.66	2.44	10.46	101	8.67E-18

Table 4.16: Fall 2018 Paired Samples Test

4.2.4 OpenDSA Analysis Result

In this section, we present the analysis result from OpenDSA exercise usage data. For the Khan Exercise (see Figure 3.6), there were 591 total attempts by students. As we discussed earlier, students had to make five correct answers to get full credit. For this reason, we expected to have at least five attempts from each student. There were 105 students who

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	post-test 1	2.04	106	0.77	0.07
	pre-test	1.62	106	0.89	0.09
Pair 2	post-test 2	1.95	103	0.78	0.08
	pre-test	1.59	103	0.90	0.09
Pair 3	post-test 3	1.85	102	0.91	0.09
	pre-test	1.58	102	0.87	0.09

Table 4.17: Fall 2018 Question 10 - 12 Paired Samples Statistics

Paired Samples Correlations				
		N	Correlation	Sig.
Pair 1	post-test 1 & pre-test	106	0.44	1.81.E-06
Pair 2	post-test 2 & pre-test	103	0.42	1.02.E-05
Pair 3	post-test 3 & pre-test	102	0.27	5.15.E-03

Table 4.18: Fall 2018 Question 10 - 12 Paired Samples Correlations

Paired Samples Test									
Pairs		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	post-test 1 - pre-test	0.42	0.88	0.09	0.25	0.59	4.90	105	3.44E-06
Pair 2	post-test 2 - pre-test	0.36	0.91	0.09	0.19	0.54	4.07	102	9.31E-05
Pair 3	post-test 3 - pre-test	0.27	1.07	0.11	0.06	0.48	2.59	101	1.10E-02

Table 4.19: Fall 2018 Question 10 - 12 Paired Samples Test

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
pre-test	73%	85%	70%	57%	97%	74%	56%	46%	82%	57%	41%	64%
post-test 1	79%	97%	98%	83%	99%	96%	67%	71%	84%	80%	60%	62%
post-test 2	81%	99%	94%	85%	100%	92%	89%	83%	95%	81%	55%	57%
post-test 3	77%	100%	96%	76%	98%	90%	96%	82%	89%	84%	46%	56%

Table 4.20: Fall 2018 Test Correctness

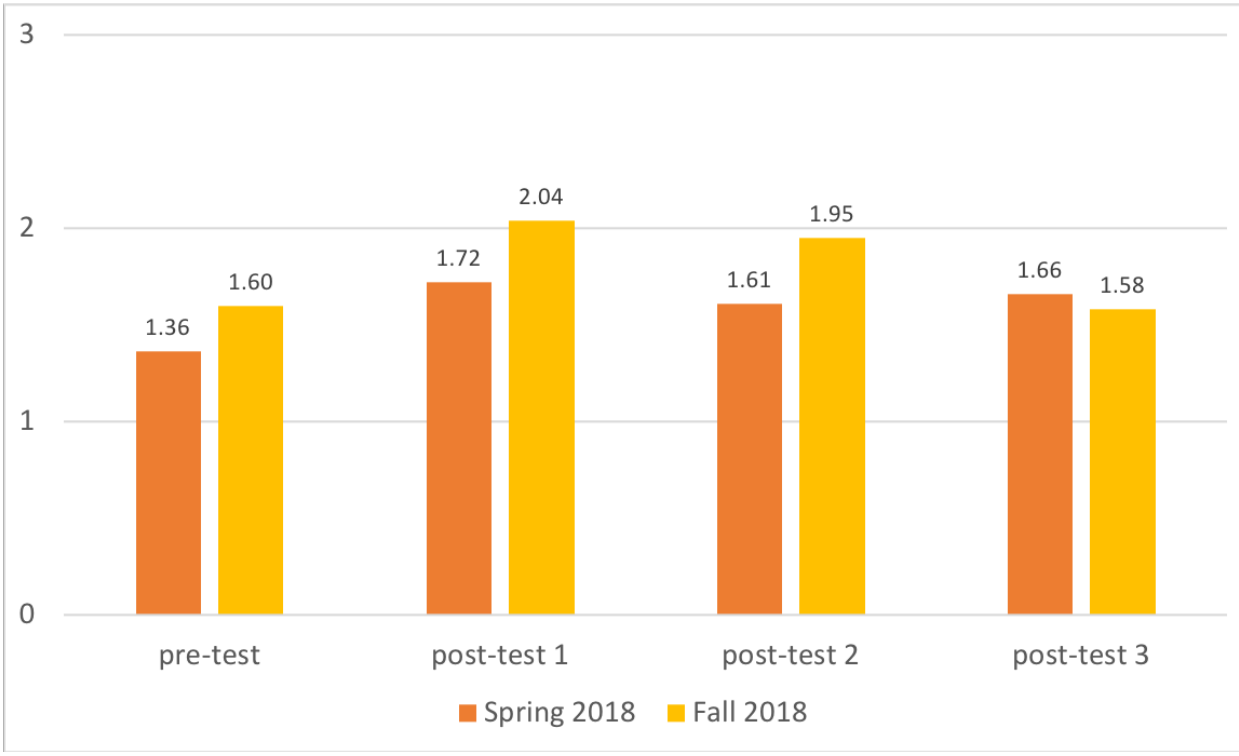

Figure 4.5: Question 10 - 12 Mean Comparison Spring 18 and Fall 18

used our Khan exercise, and the average number of trials was 5.67 times. There were 45 students (43% of the class) who used the exercise more than 5 times. There was no one who used the exercise after they receive full credit (see Section 3.2), so we did not see effective reuse of the Khan Exercise.

A total of 119 students used the JSAV exercise (see Figure 3.7) and we had 183 total attempts. Most of them only used it once to get credit, as we explained in Section 3.2. The mean time taken for this exercise was 1 minute and 35 seconds to receive full credit. There were 32 re-trials by students who tried more than 5 attempts, which was 27% of the class. They were able to answer correctly within one or two more attempts. There were 8 students who used the JSAV Exercise more than once at least a few hours later than their first usage. One of them had to retry the exercise to get the full credit, so we can say only 7 students came back to reuse it voluntarily. The average time they come back to reuse was 18 hours later, and no one used again after 24 hours from the first attempt. Students who reused the JSAV exercise voluntarily showed higher knowledge gain on Questions 10 - 12, which we targeted to improved knowledge gain from this exercise, than the class mean. We cannot conclude that reusing JSAV helped students for higher knowledge gain, because of the small number to compare. However, it showed a positive effect which we should continue to study in the future with more participants. Figure 4.6 shows the comparison between JSAV Exercise users and the overall class in Spring 2018 and Fall 2018.

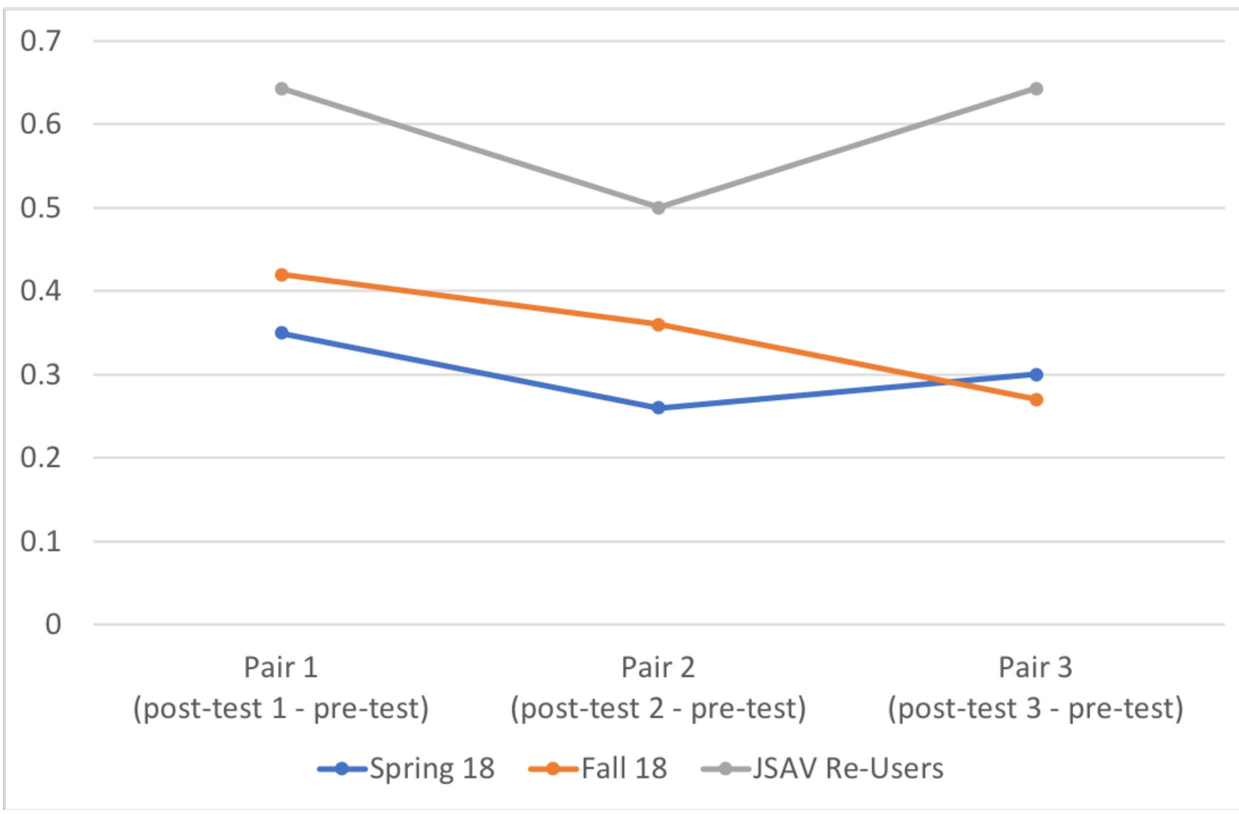

Figure 4.6: Question 10 - 12 Mean Comparison Spring 18 and Fall 18

4.2.5 Canvas Data Analysis

We also analysed students' canvas usage data to see the relationship between usage amount and their knowledge gain. The reading assignment page was used 198 times in Fall 2018. A total OF 156 attempts (79%) of the total, were made before the week of Day 8, which shows that the majority students were already influenced before taking the pre-test.

4.2.6 Survey

	Question		Count	%
1	Do you recall the interactive animations?	Yes	83	73%
		No	30	27%
2	What was the helpfulness of the interactive animations in your learning of the concept of iteration?	Very Helpful	8	7%
		Helpful	55	47%
		Neither	43	36%
		Unhelpful	6	5%
		Very Unhelpful	1	1%
3	Did you use the interactive animations after your first encounter with them?	>2	12	11%
		1-2	44	39%
		0	57	50%
	Total	113		

Table 4.21: Fall 2018 Survey Result

On the last day of the Fall 2018 class (Day 29), we conducted a survey to ask about various concepts (see Figure 3.15). 118 students completed the survey. The first question asked if the students remembered the interactive visualization. About 73% students answered yes, which means a majority of the students remember that the interactive visualization was used. The second question asked if students felt the interactive visualization was helpful and the response was positive. About 53% of the students answered that our interactive visualization was either helpful or very helpful. Only 6% of students answered the question stating that the interactive visualization was unhelpful or very unhelpful. In our data analysis, by calculating differences between the post-test 3 result and the pre-test result, students who answered with the choice helpful and very helpful showed average 15.25% knowledge gain. However, students who answered unhelpful and very unhelpful showed average 9.72% knowledge gain.

For the last question, we wanted to check if students are reusing the visualization. Only 2 out of these 7 students who answered that interactive visualization were unhelpful used the interactive visualization more than once. On the other hand, 70% of students who answered positively also answered that they reused the visualization. The survey showed that about 50% of the students did not reuse it at all. There were not a significant difference in learning gains between students who reused vs not reused.

4.3 Spring 2019

4.3.1 Spring 2019 CT Class

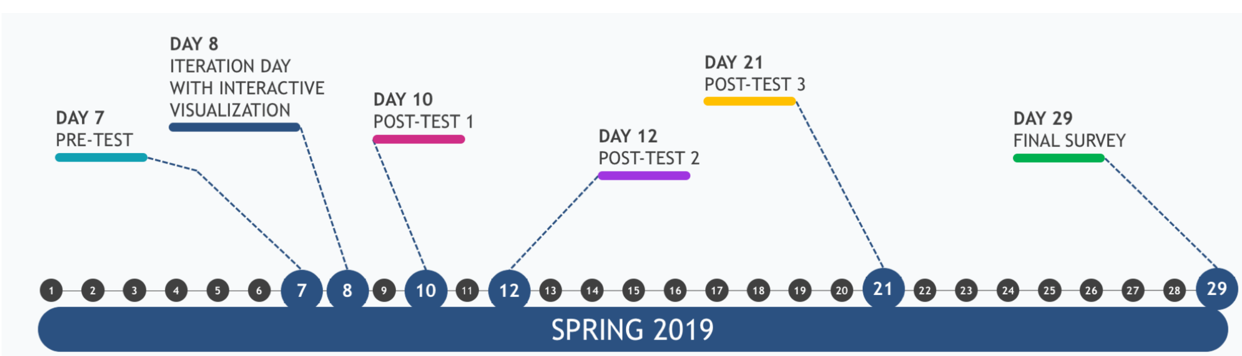

Figure 4.7: CT Course Timeline on Spring 2019

In Spring 2019, we conducted the final set of pre-post tests before and after Day 8 of the

CT class (see Figure 4.7). We conducted the pre-test before Day 8 (February 19th, 2019) with the same interactive visualization that we used for the Fall 2018 classes. Most of the conditions were the same as Spring 2018 and Fall 2018, but we made sure to conduct the pre-test well before Day 8 so that students have not seen the iteration materials yet. One section was taught by Dr. Kafura and the other section was taught by Doctoral student Javier Tibau. Like Fall 2018, the roster showed that 45% of the students were STEM related majors, which was much higher than Spring 2018. Spring semester included students from a variety of majors as summarized in Table 4.22 and 4.23. The roster data showed a lot more General Engineering students than in Spring 2018. We had 19% Freshmen, 36% Sophomores, 23% Juniors and 23% Seniors. No honors students participated. (see Table 4.22).

		Section 1		**Section 2**		**Total**	
		Count	%	Count	%	Count	%
Major	STEM	21	41%	9	27%	38	45%
	Other	30	59%	24	73%	46	55%
Year	Freshman	9	18%	7	21%	16	19%
	Sophomore	19	37%	11	33%	30	36%
	Junior	13	25%	6	18%	19	23%
	Senior	10	20%	9	27%	19	23%
Honor Students		0	0%	0	0%	0	0%
Number of Students		51		33		84	

Table 4.22: Spring 2019 Class Demographics

Table 4.23: Spring 2019 Major Demographics

Major	**Count**	**%**	STEM
Art (Fine Arts)	2	2%	
Biological Sciences	6	7%	YES
Building Construction	16	19%	
Business Information Tech	2	2%	YES
Chemistry	1	1%	YES

Table 4.23 – continued from previous page

Major	Count	%	STEM
Communication Studies	1	1%	
Comp Model & Data Anyl	3	4%	YES
Construction Engineering & Mgt	1	1%	
Electrical Engineering	1	1%	YES
Enviro Policy & Planning	2	2%	
Explore Technology	1	1%	YES
Fashion Merch. and Design	1	1%	
Finance	2	2%	
General Engineering	12	14%	YES
Geography	1	1%	
History	1	1%	
Human Development	3	4%	
Mathematics	1	1%	YES
Multimedia Journalism	1	1%	
Music	1	1%	
Nanoscience	1	1%	YES
Nat's Security & Foreign Aff	2	2%	
Packaging Systems & Design	1	1%	
Political Science	3	4%	
Prof & Tech Writing	2	2%	
Psychology	8	10%	YES
Resid Envir & Design	1	1%	
Statistics	1	1%	YES

Table 4.23 – continued from previous page

Major	Count	%	STEM
University Studies	6	7%	
Total	84		21

4.3.2 Participants

	Signed Consent Form	Participants
pre-test	76	76
post-test 1	67	67
post-test 2	70	70
post-test 3	66	73
Total	87	97

Table 4.24: Spring 2019 Participants Demographics

Three paired-sample t-tests were conducted to compare the knowledge level of iteration before and after learning iteration material on Day 8. 76 students participated in the pre-test, 67 students participated Post-test 1, 70 students Participated post-test 2, and 73 students Participated post-test 3 (see Table 4.24). In this semester, there were fewer participants than other semesters. The class was run the same way and there were no unusual events. Students who did not participate were either absent or did not sign the consent form [21].

4.3.3 Pre-Post Test Results

In Spring 2019, we conducted the pre-test on Day 7, Post-test 1 on Day 10, Post-test 2 on Day 12, and Post-test 3 on Day 21 (see Figure 4.7). Students were asked to go to the

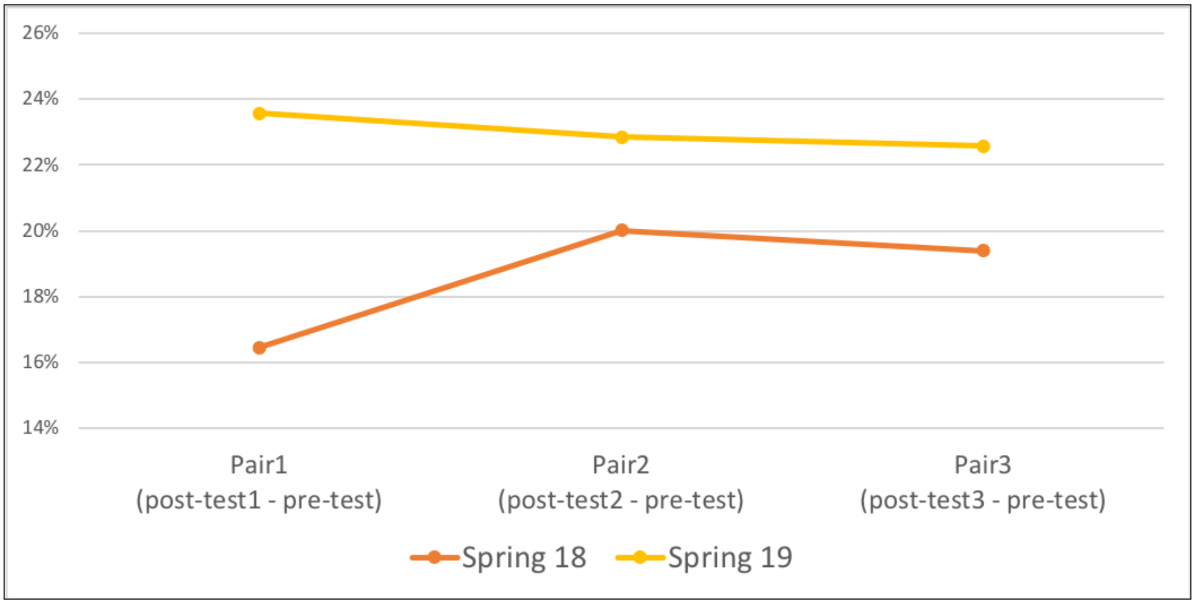

Figure 4.8: Knowledge Gain Comparison

CT Course Canvas page and complete the tests at the beginning of the classes within 20 minutes. We had the same structure and questions on the pre-test and post-tests as Spring 2018 and Fall 2018 to make the conditions equal. The full set of the questions asked can be found in Appendix A. Three paired-sample t-tests were conducted to compare the knowledge level of iteration before and after learning iteration material on Day 8. When we say that a result is significant, we mean that the p-value is less than 0.05. There was a significant difference in the scores between the pre-test (M=6.77, SD=3.17) and Post-test 1 (M=9.59, SD=1.88); t(63)=9.32, p =1.80E-13. Also, There was a significant difference in the scores between the pre-test (M=6.79, SD=3.14) and Post-test 2 (M=9.53, SD=11.38); t(65)=8.88, p =8.01E-13. There was a significant difference in the scores between the pre-test (M=6.54, SD=3.14) and Post-test 3 (M=9.25, SD=2.49); t(61)=8.57, p =4.56E-12. Tables 4.25, 4.26, and 4.27 show the full set of T-Test results for Spring 2019.

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	post-test 1	9.59	64	1.88	0.23
	pre-test	6.77	64	3.17	0.40
Pair 2	post-test 2	9.53	66	2.09	0.26
	pre-test	6.79	66	3.14	0.39
Pair 3	post-test 3	9.25	62	2.49	0.32
	pre-test	6.54	62	3.14	0.40

Table 4.25: Spring 2019 Paired Samples Statistics

Paired Samples Correlations				
		N	Correlation	Sig.
Pair 1	post-test 1 & pre-test	64	0.65	8.33E-09
Pair 2	post-test 2 & pre-test	66	0.60	8.33E-08
Pair 3	post-test 3 & pre-test	62	0.63	4.05E-08

Table 4.26: Spring 2018 Paired Samples Correlations

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	post-test1 - pre-test	1.98	1.75	0.17	1.63	2.32	11.41	101	7.129E-20
Pair 2	post-test2 - pre-test	2.40	1.79	0.18	2.05	2.75	13.56	101	1.737E-24
Pair 3	post-tet3 - pre-test	2.33	1.92	0.20	1.93	2.72	11.67	92	7.459E-20

Table 4.27: Spring 2018 Paired Samples Test

We had 7% STEM majors in Spring 2018 and 25% STEM majors in Spring 2019. The pre-test scores between Spring 2018 and Spring 2019 were similar. Because of this, we assume that the fraction of STEM majors did not influence the pre-knowledge level for iteration. We call the pre-test and Post-test 1 as Pair 1. We call the pre-test and Post-test 2 as Pair 2. We call the pre-test and Post-test 3 as Pair 3. As Figure 4.8 shows, all pairs of Spring 2019

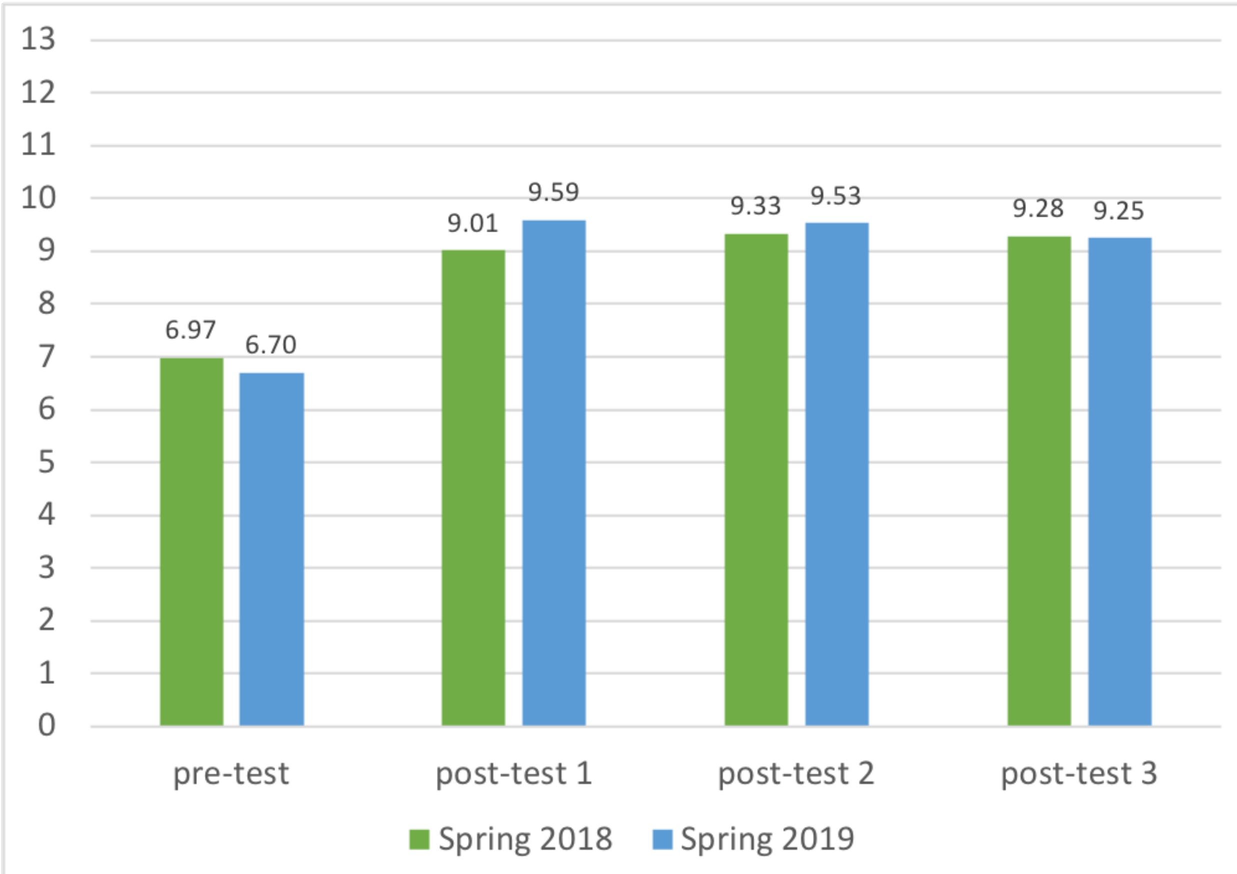

Figure 4.9: Mean comparison between Spring 18 and Spring 19

showed a greater knowledge gain than Spring 2018. In Pair 1, Spring 2019 (M = 2.83, SD = 2.43) showed greater knowledge gain than Spring 2018 (M = 1.98, SD = 1.75), d =0.40. In Pair 2, Spring 2019 (M = 2.74, SD = 2.51) showed greater knowledge gain than Spring 2018 (M = 2.40, SD = 1.79), d =0.16. In Pair 3, Spring 2019 (M = 2.83, SD = 2.43) showed greater knowledge gain than Spring 2018 (M = 1.98, SD = 1.75), d =0.17.

Also, three more paired-sample t-tests were conducted to compare the knowledge level of iteration concepts covered in Questions 10 - 12 (see Table 3.1) before and after iteration material was presented on Day 8. There was a significant difference in the scores between the pre-test (M=1.42, SD=1.01) and Post-test 1 (M=2.13, SD=0.81); t(63)=5.65, p =4.08E-07. There was a significant difference in the scores between the pre-test (M=1.41, SD=1.00)

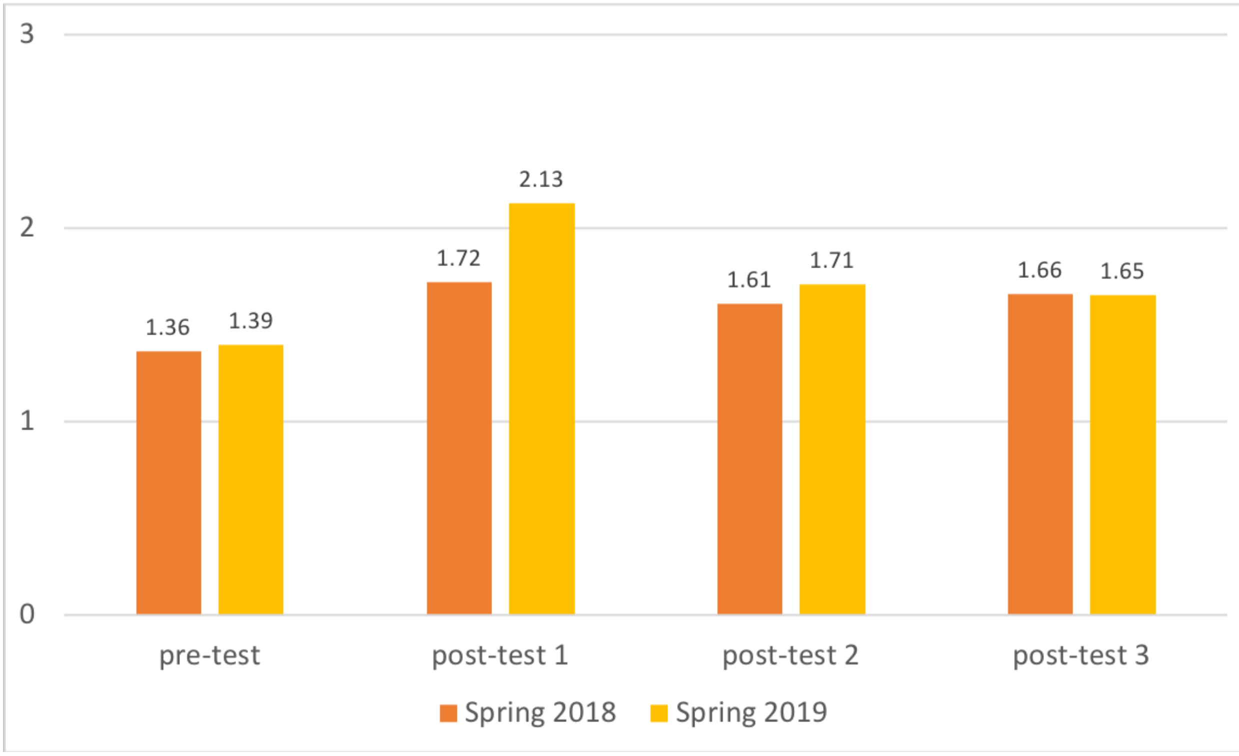

Figure 4.10: Questions 10 - 12 mean comparison between Spring 18 and Spring 19

and Post-test 2 (M=1.71, SD=0.91); t(65)=2.55, p = 1.30E-02. There was a significant difference in the scores between the pre-test (M=1.35, SD=1.01) and Post-test 3 (M=1.65, SD=0.89); t(61)=2.45, p =1.72E-02. Tables 4.28, 4.29, and 4.30 show the full set of T-Test results for Question 10 - 12. As Figure 4.8 shows, Pair 1 of Spring 2019 (M = 0.70, SD = 0.99) showed greater knowledge gain than Spring 2018 (M = 0.35, SD = 0.90), d = 0.37. This result shows that students showed greater knowledge gain of the concepts that Question 10 - 12 covered right after Day 8, but showed similar knowledge gain in Post-test 2 and Post-test 3.

We also analysed the correctness of each question (see Table 4.31). We found out that the correctness increased more in Spring 2019 than Spring 2018, except for Question 4 (see Figure 4.11). Question 4 is asking about the concept of sum in the body of the iteration. (see Table 3.1 and Appendix A). The pre-test result of Question 4 in Spring 2018 was very low, but the result of the post-tests for the question were similar between Spring 2018 and 2019.

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	post-test 1	2.13	64	0.81	0.10
	pre-test	1.42	64	1.01	0.13
Pair 2	post-test 2	1.71	66	0.91	0.11
	pre-test	1.41	66	1.00	0.12
Pair 3	post-test 3	1.65	62	0.89	0.11
	pre-test	1.35	62	1.01	0.13

Table 4.28: Spring 2019 Question 10 - 12 Paired Samples Statistics

Paired Samples Correlations				
		N	Correlation	Sig.
Pair 1	post-test 1 & pre-test	64	0.42	5.47.E-04
Pair 2	post-test 2 & pre-test	66	0.49	2.48.E-05
Pair 3	post-test 3 & pre-test	62	0.47	1.35.E-04

Table 4.29: Spring 2018 Question 10 - 12 Paired Samples Correlations

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	post-test 1 - pre-test	0.70	0.99	0.12	0.45	0.95	5.65	63	4.08.E-07
Pair 2	post-test 2 - pre-test	0.30	0.96	0.12	0.07	0.54	2.55	65	1.30.E-02
Pair 3	post-test 3 - pre-test	0.31	0.98	0.13	0.06	0.56	2.45	61	1.72.E-02

Table 4.30: Spring 2018 Question 10 - 12 Paired Samples Test

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
pre-test	56%	85%	47%	45%	84%	63%	48%	44%	61%	51%	33%	53%
post-test 1	78%	99%	91%	78%	99%	87%	75%	70%	78%	77%	67%	72%
post-test 2	77%	99%	93%	81%	99%	84%	86%	79%	89%	76%	41%	56%
post-test 3	71%	94%	91%	71%	97%	88%	88%	77%	83%	73%	29%	62%

Table 4.31: Spring 2019 Test Correctness

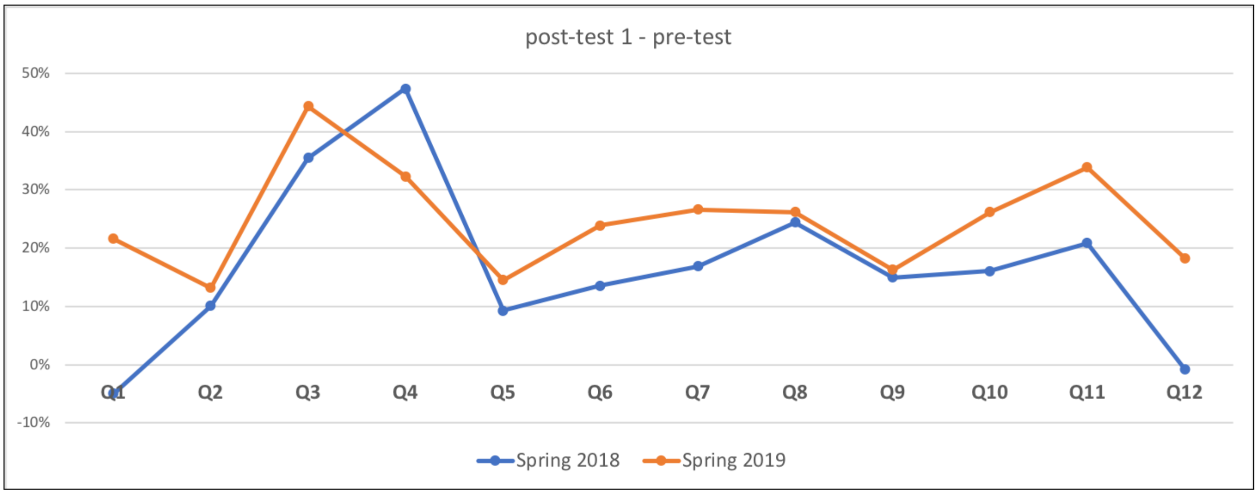

Figure 4.11: Correctness Gain for Each Quiz Question

4.3.4 OpenDSA Analysis Result

For Spring 2019, 80 students used the JSAV exercise and we had 108 total attempts. Most of them used it one time, to get full credit, as we explained in Section 3.2. The mean time taken for this exercise was 121 seconds. 23 students used JSAV exercises more than one time, but most of them did so within one hour after they started the first attempt. Average number of re-uses was 1.33 times. There were 23 re-trials by students who missed the first attempt. They were able to answer correctly within 2.17 attempts. There were only two students who used the JSAV exercise more than once at least a few hours later, and one of them was to retry the exercise and get full credit, so we can say only one student came back. The student reused the JSAV exercise 22 hours later and no one used it again after 24 hours from the first attempt.

For the Khan Exercise (see Figure 3.6), there were 475 total attempts from students. Students had to give 5 correct answers to get full credit. For this reason, we expected to have at least 5 attempts from the students. There were 73 students who used our Khan exercise, and the average number of uses was 5.46. There were 40 students who used it more than 5 times, which is 55% of the group who used the Khan Exercise. Students who used it more than 5 times had an average of 6.7 total attempts. There was no one who used the Khan exercise after they receive credit.

4.3.5 Canvas Data Analysis

We also analyze students' canvas usage data to see the relationship between usage amount and the knowledge gain. We collected and analyzed Canvas usage from 66 out 97 students (68%), after they signed the consent form. There were total of 99 accesses from the students. A total of 66 attempts were made before the week of Day 8, 19 were made on the week of Day 8, and 12 attempts after the week of Day 8. There were 25 students who used it again at least a few hours after the first usage.

4.3.6 Survey

On the last day of the Spring 2019 class (Day 29), 83 students completed a survey (see Figure 3.15). The first question asked if the students remembered the interactive visualization. About 73% students answered yes, which means a majority of the students remember that the interactive visualization was used. The second question asked if students felt the interactive visualization was helpful and the response was positive. About 52% of the students answered that our interactive visualization was either helpful or very helpful. Only 8% of

	Question		Count	%
1	Do you recall the interactive animations?	Yes	83	71%
		No	24	29%
2	What was the helpfulness of the interactive animations in your learning of the concept of iteration?	Very Helpful	8	10%
		Helpful	35	42%
		Neither	33	40%
		Unhelpful	5	6%
		Very Unhelpful	2	2%
3	Did you use the interactive animations after your first encounter with them?	>2	9	11%
		1-2	27	33%
		0	47	57%
	Total	83		

Table 4.32: Spring 2019 Survey Result

students answered the question stating that the interactive visualization was unhelpful or very unhelpful. For the last question, we wanted to check if students are reusing the visualization. The survey showed that about 57% of the students did not reuse it at all, about 33% of students used it once or twice, and about 11% of students used it more than twice.

4.4 Summary

We have conducted pre-post tests to compare knowledge gain differences, and how theses relate to the effectiveness of using interactive visualizations and exercises in CT courses in Spring 2018, Fall 2018, and Spring 2019. In Fall 2018, students used interactive visualizations and exercises for learning iteration. The students showed higher grades on the post-test than in Spring 2018 where we did not use the interactive visualizations and exercises. The "pre-test" for Fall 2018 turned out to be unreliable as an indicator of prior knowledge, because some students accessed the reading assignment before the pre-test, so we could not claim students showed higher knowledge gain by using visualizations compared to Spring 2018. In Spring 2019, we conducted the pre-test with the same condition as the pre-test in Spring 2018. Even though we a had larger pool of STEM majors in Spring 2019, the pre-test showed similar scores as in Spring 2018, so a student's major was not a significant factor in their basic knowledge level of iteration. Spring 2019 students, who used interactive visualizations, showed higher knowledge gain than students in Spring 2018. However, the absolute post-test scores were not high enough to be statistically significant. So we cannot say that our interactive visualizations and exercises helped for learning iteration. However, we found some positive results from analysing usage data and the Survey. The number of students was small, but these students who used the JSAV exercise more than once showed better scores on Question 10-12 than the class mean. Also, survey responses were positive about their recall and helpfulness. Students who answered that the visualizations were "helpful" or "very helpful" showed higher scores than the class mean. In the future, we are planning to update more features of our visualizations and system. We hypothesize that students will show higher scores and higher knowledge gain of iteration concepts when using our new interactive visualizations and exercises than in previous semesters.

Chapter 5

Threats to Validity

There are four threats to the validity of this experiment. First, the pre-test in Fall 2018 was delivered on Day 8 of the class, not Day 7, which was different than for our control group. We had the test at the beginning of the class, which is before students were given in-class instruction about lists and iteration. However, students were expected and recommended to complete the reading assignment before they attended the class. Thus, the students may have already gained some understanding of the material prior to the pre-test. For this reason, we cannot say that the pre-test result was conducted in the same way as Spring 2018. This concern explains why the "pre-test" scores was so high in Fall 2018. Due to this, we made the decision to run pre-post tests again in Spring 2019. At this time, we conducted the test on Day 7, like Spring 2018, to make sure to measure students' pre-knowledge level before they experience in interactive visualization or any other reading material related with iteration concept. We see that the pre-test scores for Spring 2019 are similar to Spring 2018. Second, each semester, the teaching staff changed. But Dr. Kafura led staff meetings each term. In Spring 2018, Dr. Kafura and Dr. Bart taught sections of the course. In Fall 2018, Dr. Kafura taught both sections. For Spring 2019, Dr. Kafura and Javier Tibau each taught one section. Even though the teaching staff changed, they used exactly the same materials including reading assignments, class activities, quizzes, etc. Also, all the faculty members for both classes had weekly meetings together so they could maintain the classes' condition equally.

For the last, we used students' major as an indicator of better ability. However, we do not know about their prior experience in computing or programming. For example, there could be a student who is a History major but had a CS background. Also, we never reported any evidence about whether STEM vs. non-STEM matters or not. We plan in future work to analyze whether these two groups have different performance outcomes or not.

Chapter 6

Conclusions

We developed new interactive visualizations using OpenDSA to assist student's in learning the concept of iteration. Based on current interactive visualization research, we hypothesized that students who used visualizations will show higher knowledge gain on the topic of iteration in a CT course. We included interactive visualization in the course material for Fall 2018 and Spring 2019. To measure the knowledge gain of students across the semesters, we conducted a pre-test and three post-tests. The results showed that students had greater knowledge gain with interactive visualization (see Figure 4.8 and 4.9). Also, student responses on the survey showed positive rating of the interactive visualizations.

We tried to conduct individual interviews with students of the CT course in both Fall 2018 and Spring 2019. However, no students volunteered to participate for this research. If we were to conduct these interviews again, we believe offering some compensation would help us to recruit participants.

For the usage data analysis, we were not able to collect click-stream data from the visualizations in Canvas, and instead only collected attempts from exercises. For this reason, we had difficulties to fully understand students' usage behavior. For example, we could not determine the number of times the students clicked the "previous" button on the visualization to go back to the previous slide, or determine the number of students who skiped the visualization entirely. We have updated the OpenDSA system to collect these data in the future.

From the Fall 2018 survey, about 50% of students answered that they did not re-use our interactive visualization. Through conducting interviews or adding questions on the Survey, we might be able to assess why some students did and others did not re-use our interactive visualizations and exercises. For some OpenDSA content, students are known to use the visualization and exercises as study aides for exams [15].

We should create variations on problem instances for the interactive exercises. At this time, we only had a question asking about the status of the value after one execution in the iteration body. As Figure 4.11 shows, for Spring 2019, students had more knowledge gain on Question 11 than Question 10. Question 11 asked about the status of the value after one execution and Question 10 was almost the same question, but it asked the status of the value before execution (see Appendix A). For this reason, we are planning to develop more exercises, which would ask about the value's change both before or after execution and also ask about other values. Also, results from Questions 2, 4, 8, and 9 in Spring 2019 showed only small knowledge gains sometimes, or even smaller than Spring 2018 (See 4.11). We should develop more visualizations or update the current design to cover more concepts (see Table 3.1).

We will update the grading system to prevent skipping or guessing [15]. We will subtract a point if a student make wrong answer to prevent guessing. Right now, students might choose any answers because they know that they have unlimited attempts. However, the new version of exercise will force students to make correct answers in a row, so students will receive full credit only when they demonstrate good understanding.

Students showed least knowledge gain for Question 12, which asked about the scope of the iteration. We can also design and add interactive visualizations or exercises that cover the scope of the iteration. We also plan to create visualizations on other content such as dictionaries.

Bibliography

[1] Blockpy youtube, https://youtu.be/rzaopqopmom.

[2] Khan academy website, https://www.khanacademy.org/.

[3] Opendsa khan, https://opendsa.readthedocs.io/en/latest/kaexercise.html.

[4] Think java.

[5] Khan academy exercise infrastructure, https://github.com/khan/khan-exercises, Dec 2018.

[6] Alice, http://www.alice.org/, 2019.

[7] Austin Cory Bart and Dennis Kafura. Blockpy interactive demo: Dual text/block python programming environment for guided practice and data science. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 639–640. ACM, 2017.

[8] Austin Cory Bart, Javier Tibau, Eli Tilevich, Clifford A Shaffer, and Dennis Kafura. Blockpy: An open access data-science environment for introductory programmers. *Computer*, 50(5):18–26, 2017.

[9] Paul Blanc and Rosamund Sutherland. Student teachers' approaches to investigative mathematics: Iterative engagement or disjointed mechanisms? In *PME CONFERENCE*, volume 2, pages 2–97. Citeseer, 1996.

[10] Nell B. Dale. Most difficult topics in cs1: Results of an online survey of educators. *SIGCSE Bull.*, 38(2):49–53, June 2006.

[11] Wanda Dann, Stephen Cooper, and Randy Pausch. Making the connection: programming with animated small world. In *ACM SIGCSE Bulletin*, volume 32, pages 41–44. ACM, 2000.

[12] John P Dougherty. Concept visualization in cs0 using alice. *Journal of Computing Sciences in Colleges*, 22(3):145–152, 2007.

[13] Michael Eagle and Tiffany Barnes. Wu's castle: Teaching arrays and loops in a game. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '08, pages 245–249, New York, NY, USA, 2008. ACM.

[14] Michael Eagle and Tiffany Barnes. Wu's castle: Teaching arrays and loops in a game. *SIGCSE Bull.*, 40(3):245–249, June 2008.

[15] Sally Hamouda Mohammed F.Farghally Clifford A.Shaffer Eric Fouh, Daniel A.Breakiron. Exploring students learning behavior with an interactive etextbook in computer science courses. *Computers in Human Behavior*, pages 478–485, 2014.

[16] Eric Fouh, Monika Akbar, and Clifford A. Shaffer. The role of visualization in computer science education. *Computers in the Schools*, 29(1-2):95–117, 2012.

[17] Eric Fouh, Ville Karavirta, Daniel A Breakiron, Sally Hamouda, Simin Hall, Thomas L Naps, and Clifford A Shaffer. Design and architecture of an interactive etextbook–the opendsa system. *Science of Computer Programming*, 88:22–40, 2014.

[18] Eric Fouh, Maoyuan Sun, and Clifford Shaffer. Opendsa: A creative commons active-ebook. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 721–721. ACM, 2012.

[19] Luke Gusukuma, Austin Cory Bart, Dennis Kafura, and Jeremy Ernst. Misconception-driven feedback: Results from an experimental study. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*, ICER '18, pages 160–168, New York, NY, USA, 2018. ACM.

[20] Dennis Kafura. Personal Communication, 2019.

[21] Dennis Kafura. Personal Communication, 2019.

[22] Dennis Kafura, Austin Cory Bart, and Bushra Chowdhury. A computational thinking course accessible to non-stem majors. *J. Comput. Sci. Coll.*, 34(2):157–163, December 2018.

[23] V. Karavirta, R. Haavisto, E. Kaila, M. Laakso, T. Rajala, and T. Salakoski. Interactive learning content for introductory computer science course using the ville exercise framework. In *2015 International Conference on Learning and Teaching in Computing and Engineering*, pages 9–16, April 2015.

[24] V. Karavirta and C. A. Shaffer. Creating engaging online learning material with the jsav javascript algorithm visualization library. *IEEE Transactions on Learning Technologies*, 9(2):171–183, April 2016.

[25] Ville Karavirta and Clifford A Shaffer. Jsav: the javascript algorithm visualization library. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, pages 159–164. ACM, 2013.

[26] Marie Olsson, Peter Mozelius, and Jonas Collin. Visualisation and gamification of e-learning and programming education. *Electronic journal of e-learning*, 13(6):441–454, 2015.

[27] Clifford A Shaffer. Opendsa: An interactive etextbook for computer science courses. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 5–5. ACM, 2016.

[28] Clifford A Shaffer, Ville Karavirta, Ari Korhonen, and Thomas L Naps. Opendsa: beginning a community active-ebook project. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, pages 112–117. ACM, 2011.

[29] Franklyn Turbak, Constance Royden, Jennifer Stephan, and Jean Herbst. Teaching recursion before loops in cs1. *Journal of Computing in Small Colleges*, 14(4):86–101, 1999.

[30] J. Zhang, M. Atay, E. R. Caldwell, and E. J. Jones. Visualizing loops using a game-like instructional module. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 448–450, July 2013.

Appendices

Appendix A

Pre-Post Test Questions

Question 1

pre-test	To iterate over the list shown below, what is the type of the iteration variable needed?
post-test 1	To iterate over the list shown below, what is the type of the iteration variable needed?
post-test 2	To iterate over the list shown below, what is the type of the iteration variable needed?
post-test 3	To iterate over the list shown below, what is the type of the iteration variable needed?

Question 2

pre-test	When using iteration to compute the sum of the values in the list ***price__list***, which of the following is the correct way to express the
post-test	When using iteration to compute the sum of the values in the list ***weight__list***, which of the following is the correct way to express the
post-test 2	When using iteration to compute the sum of the values in the list ***calorie__list***, which of the following is the correct way to express the
post-test 3	When using iteration to compute the sum of the values in the list ***rent__list***, which of the following is the correct way to express the

Question 3

pre-test	When using ***calorie__sum*** to add up the values in ***calorie__list***, which of the following is the correct initialization for ***calorie__sum***?
post-test 1	When using ***price__sum*** to add up the values in ***price__list***, which of the following is the correct initialization for ***price__sum***?
post-test 2	When using ***weight__sum*** to add up the values in ***weight__list***, which of the following is the correct initialization for ***weight__sum***?
post-test 3	When using ***cost__su*** to add up the values in ***cost__list***, which of the following is the correct initialization for ***cost__sum***?

Question 4

pre-test	When using the iteration shown below to compute the sum of the numbers in the ***rent_list***, which of the following is the correct statement to be in the body of the iteration?
post-test 1	When using the iteration shown below to compute the sum of the numbers in the ***distance_list***, which of the following is the correct statement to be in the body of the iteration?
post-test 2	When using the iteration shown below to compute the sum of the numbers in the ***cost_list***, which of the following is the correct statement to be in the body of the iteration?
post-test 3	When using the iteration shown below to compute the sum of the numbers in the ***calorie_list***, which of the following is the correct statement to be in the body of the iteration?

Question 5

pre-test	In the iteration shown below which of the following best describes the relationship between ***distance and distance_list***?
post-test 1	In the iteration shown below which of the following best describes the relationship between ***cost and cost_list?***
post-test 2	In the iteration shown below which of the following best describes the relationship between ***rent and rent_list?***
post-test 3	In the iteration shown below which of the following best describes the relationship between ***people and people_list?***

Question 6

pre-test	Which of the following algorithms computes the number of ***ages in age_list*** that are greater than ***20***?
post-test 1	Which of the following algorithms computes the number of ***heights in height_list*** that are greater than ***60***?
post-test 2	Which of the following algorithms computes the number of ***weights in weight_list*** that are greater than ***155***?
post-test 3	Which of the following algorithms computes the number of ***wages in wage_list*** that are greater than ***20***?

Question 7

pre-test	When using an iteration to generate a new list of integer values, which of the following is the correct initialization for the variable new_list?
post-test 1	When using an iteration to generate a new list of float values, which of the following is the correct initialization for the variable new_list?
post-test 2	When using an iteration to generate a new list of string values, which of the following is the correct initialization for the variable new_list?
post-test 3	When using an iteration to generate a new list of float values, which of the following is the correct initialization for the variable new_list?

Question 8

pre-test	Which of the following algorithms computes a new list containing all the ***ages in the list age_list*** that are greater than ***20***?
post-test 1	Which of the following algorithms computes a new list containing all the ***heights in the list height_list*** that are greater than ***60***?
post-test 2	Which of the following algorithms computes a new list containing all the ***weights in the list weight_list*** that are greater than ***155***?
post-test 3	Which of the following algorithms computes a new list containing all the ***wages in the list wage_list*** that are greater than ***20***?

Question 9

pre-test	When using the iteration shown below to compute a new list each of whose values are ***12 times the values in list (e.g., converting 5 years to 60 months)***, which of the following is the correct statement to be in the body of the iteration:
post-test 1	When using the iteration shown below to compute a new list each of whose values are ***12 times the values in list (e.g., converting 5 feet to 60 inches)***, which of the following is the correct statement to be in the body of the iteration:
post-test 2	When using the iteration shown below to compute a new list each of whose values are ***7 times the values in list (e.g., converting 7 touchdowns to 49 points)***, which of the following is the correct statement to be in the body of the iteration:
post-test 3	When using the iteration shown below to compute a new list each of whose values are ***100 times the values in list (e.g., converting 5 centuries to 500 years)***, which of the following is the correct statement to be in the body of the iteration:

Question 10

pre-test	Given the iteration shown below, what is the state immediately before the first execution of the statement indicated by the arrow?
post-test 1	Given the iteration shown below, what is the state immediately before the first execution of the statement indicated by the arrow?
post-test 2	Given the iteration shown below, what is the state immediately before the first execution of the statement indicated by the arrow?
post-test 3	Given the iteration shown below, what is the state immediately before the first execution of the statement indicated by the arrow?

Question 11

pre-test	Given the iteration and the partial state shown below, what is the value of the variable sum immediately after the execution of the statement indicated by the arrow?
post-test 1	Given the iteration and the partial state shown below, what is the value of the variable sum immediately after the execution of the statement indicated by the arrow?
post-test 2	Given the iteration and the partial state shown below, what is the value of the variable sum immediately after the execution of the statement indicated by the arrow?
post-test 3	Given the iteration and the partial state shown below, what is the value of the variable sum immediately after the execution of the statement indicated by the arrow?

Question 12

pre-test	For the following algorithm, choose the answer below that best describes how many times the print statement is executed.
post-test 1	For the following algorithm, choose the answer below that best describes how many times the print statement is executed.
post-test 2	For the following algorithm, choose the answer below that best describes how many times the print statement is executed.
post-test 3	For the following algorithm, choose the answer below that best describes how many times the print statement is executed.

Appendix B

Pre-Post Result Data

B.1 Spring 2018

Table B.1: Spring 2018 T-Test Data

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student001	#N/A	6	#N/A	#N/A
Student002	7	5	10	10
Student003	#N/A	4.5	9.5	9
Student004	9.5	9	9.5	8.5
Student005	7.5	8.5	9	#N/A
Student006	5.5	8	8	#N/A
Student007	#N/A	8.5	8.5	#N/A
Student008	10.5	11	11	11
Student009	8.5	10.5	11.5	10.5
Student010	10.5	11	11	#N/A
Student011	6	10.5	9	9.5
Student012	4.5	#N/A	#N/A	#N/A
Student013	8.5	10	9	10
Student014	9	10.5	6.5	8.5

Table B.1 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student015	9	8.5	8	9.5
Student016	9	9.5	#N/A	#N/A
Student017	3	4	4	7
Student018	10.5	11	10.5	11
Student019	#N/A	10	#N/A	#N/A
Student020	#N/A	8	6	5.5
Student021	5.5	7.5	6.5	8.5
Student022	6	10	9	8
Student023	12	11	12	12
Student024	7.5	#N/A	#N/A	#N/A
Student025	3.5	6.5	5.5	6.5
Student026	7.5	10.5	10.5	11
Student027	8	#N/A	8.5	10.5
Student028	7	9.5	10	7.5
Student029	8.5	10.5	10.5	10.5
Student030	8.5	10.5	8.5	9
Student031	7	11	10	9
Student032	#N/A	#N/A	9	10
Student033	6	#N/A	#N/A	#N/A
Student034	5.5	6.5	10.5	9
Student035	8.5	11	10	10
Student036	7	7	9.5	8.5
Student037	9.5	12	11	11

Table B.1 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student038	#N/A	8	9.5	#N/A
Student039	2	7	6	9
Student040	6	11	11	10
Student041	7	10	12	#N/A
Student042	7.5	7.5	#N/A	#N/A
Student043	6	8	10	9
Student044	10	11	11	11
Student045	7.5	11	10	9.5
Student046	6	8.5	10	11
Student047	#N/A	6.5	9	9
Student048	8	10	12	12
Student049	9	11	12	10
Student050	9	10	11	12
Student051	12	12	12	#N/A
Student052	9	10.5	#N/A	#N/A
Student053	9.5	9	8.5	#N/A
Student054	7.5	10.5	8.5	10
Student055	9	#N/A	12	7
Student056	3	6	8	#N/A
Student057	7.5	10	11	11
Student058	9.5	10.5	11.5	12
Student059	5.5	11.5	10	9
Student060	5.5	7.5	10	10

Table B.1 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student061	8.5	12	12	11
Student062	1	6	5.5	#N/A
Student063	4.5	7.5	7.5	9.5
Student064	7.5	7.5	6.5	8.5
Student065	5	6	9	10
Student066	7	12	11	11
Student067	#N/A	#N/A	8	4
Student068	4.5	9	9	9
Student069	2.5	5.5	7	9
Student070	6.5	7.5	8.5	8.5
Student071	4.5	7.5	9.5	6.5
Student072	4.5	5.5	7.5	4
Student073	4	8	9.5	4.5
Student074	7	9.5	7.5	9
Student075	7.5	9.5	9.5	9.5
Student076	4	5	8	6.5
Student077	3	4	4	6
Student078	6.5	8.5	8.5	8.5
Student079	12	10.5	10.5	8.5
Student080	7	12	10	12
Student081	10	10	11	11
Student082	4	7	8	8.5
Student083	6	8.5	9.5	8

Table B.1 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student084	5	11.5	11	10
Student085	7.5	10.5	11	9.5
Student086	10.5	10.5	11.5	9.5
Student087	8.5	10.5	#N/A	8.5
Student088	9.5	10	11	10
Student089	10.5	10.5	11.5	11
Student090	10	12	12	11
Student091	4	8.5	9.5	#N/A
Student092	6	#N/A	9	10.5
Student093	5	7	9	10
Student094	11	10	11	11
Student095	5	8	9.5	8.5
Student096	6.5	8.5	9.5	9.5
Student097	8.5	10.5	10	11
Student098	#N/A	9.5	7.5	10
Student099	11	11	11	12
Student100	7.5	11	12	10
Student101	6	6	7.5	8
Student102	7.5	9	10.5	12
Student103	6	8.5	9.5	5
Student104	5.5	5	7	9.5
Student105	6	6.5	6.5	#N/A
Student106	2.5	6	6.5	7

Table B.1 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student107	6.5	8	9	8
Student108	1.5	4.5	6	5
Student109	10.5	11	10.5	11
Student110	10	12	12	11
Student111	7	11	11	10
Student112	7.5	7.5	#N/A	#N/A
Student113	7.5	11	11.5	11
Student114	4.5	8.5	#N/A	8.5
Student115	8	#N/A	10	11
Student116	7	7	8.5	#N/A
Student117	2	8	6.5	8.5
Student118	10	12	10	11
Student119	4	#N/A	#N/A	#N/A
Student120	6	6	6.5	7.5
Student121	2.5	#N/A	5	7.5
Student122	2	#N/A	5	2

Table B.2: Spring 2018 T-Test Data, Question 10 - 12

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student001	#N/A	2	#N/A	#N/A
Student002	0	1	2	1

Table B.2 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student003	#N/A	0.5	1.5	2
Student004	2.5	2	1.5	1.5
Student005	0.5	1.5	1	#N/A
Student006	0.5	2	1	#N/A
Student007	#N/A	1.5	0.5	#N/A
Student008	2.5	3	3	3
Student009	2.5	2.5	2.5	1.5
Student010	1.5	2	2	#N/A
Student011	1	2.5	1	1.5
Student012	1.5	#N/A	#N/A	#N/A
Student013	0.5	1	1	1
Student014	2	2.5	0.5	1.5
Student015	2	0.5	0	1.5
Student016	1	2.5	#N/A	#N/A
Student017	1	0	1	1
Student018	2.5	3	1.5	2
Student019	#N/A	2	#N/A	#N/A
Student020	#N/A	2	1	0.5
Student021	1.5	1.5	1.5	0.5
Student022	0	2	0	2
Student023	3	2	3	3
Student024	1.5	#N/A	#N/A	#N/A
Student025	1.5	1.5	1.5	1.5

Table B.2 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student026	1.5	1.5	1.5	2
Student027	2	#N/A	1.5	1.5
Student028	3	2.5	2	1.5
Student029	1.5	1.5	1.5	1.5
Student030	2.5	2.5	1.5	1
Student031	2	3	3	2
Student032	#N/A	#N/A	2	2
Student033	1	#N/A	#N/A	#N/A
Student034	1.5	1.5	1.5	2
Student035	1.5	2	2	2
Student036	2	1	2.5	1.5
Student037	1.5	3	2	3
Student038	#N/A	1	1.5	#N/A
Student039	0	0	0	1
Student040	1	2	2	2
Student041	2	2	3	#N/A
Student042	1.5	1.5	#N/A	#N/A
Student043	0	1	1	1
Student044	2	2	2	2
Student045	1.5	3	2	1.5
Student046	1	0.5	2	2
Student047	#N/A	1.5	2	3
Student048	2	3	3	3

Table B.2 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student049	2	3	3	2
Student050	2	1	2	3
Student051	3	3	3	#N/A
Student052	1	2.5	#N/A	#N/A
Student053	1.5	2	1.5	#N/A
Student054	1.5	1.5	0.5	2
Student055	1	#N/A	3	1
Student056	1	1	1	#N/A
Student057	2.5	2	3	2
Student058	1.5	1.5	2.5	3
Student059	1.5	2.5	2	3
Student060	1.5	0.5	2	2
Student061	1.5	3	3	3
Student062	0	1	1.5	#N/A
Student063	0.5	0.5	0.5	0.5
Student064	2.5	1.5	1.5	1.5
Student065	1	2	1	1
Student066	2	3	2	3
Student067	#N/A	#N/A	1	2
Student068	0.5	1	1	2
Student069	0.5	0.5	2	1
Student070	0.5	1.5	0.5	0.5
Student071	0.5	0.5	0.5	0.5

Table B.2 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student072	1.5	0.5	1.5	0
Student073	0	1	1.5	1.5
Student074	2	1.5	0.5	1
Student075	1.5	1.5	1.5	1.5
Student076	0	0	0	0.5
Student077	0	0	0	0
Student078	0.5	0.5	0.5	1.5
Student079	3	1.5	1.5	0.5
Student080	1	3	2	3
Student081	2	2	2	2
Student082	0	0	0	0.5
Student083	1	1.5	0.5	1
Student084	1	2.5	2	2
Student085	0.5	1.5	2	1.5
Student086	1.5	1.5	2.5	1.5
Student087	2.5	2.5	#N/A	1.5
Student088	1.5	2	2	3
Student089	1.5	1.5	2.5	2
Student090	1	3	3	2
Student091	0	2.5	1.5	#N/A
Student092	1	#N/A	1	1.5
Student093	0	1	1	1
Student094	3	2	2	2

Table B.2 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student095	0	0	1.5	1.5
Student096	0.5	1.5	1.5	1.5
Student097	2.5	1.5	1	2
Student098	#N/A	0.5	0.5	2
Student099	2	3	2	3
Student100	0.5	3	3	3
Student101	2	2	1.5	1
Student102	1.5	2	1.5	3
Student103	1	1.5	1.5	0
Student104	0.5	1	1	1.5
Student105	3	0.5	0.5	#N/A
Student106	0.5	0	1.5	0
Student107	0.5	2	1	2
Student108	1.5	1.5	2	1
Student109	1.5	2	1.5	2
Student110	3	3	3	3
Student111	3	3	3	2
Student112	1.5	1.5	#N/A	#N/A
Student113	1.5	3	2.5	2
Student114	1.5	1.5	#N/A	1.5
Student115	1	#N/A	2	2
Student116	1	1	1.5	#N/A
Student117	0	2	1.5	1.5

Table B.2 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student118	3	3	1	3
Student119	1	#N/A	#N/A	#N/A
Student120	0	1	1.5	1.5
Student121	1.5	#N/A	1	1.5
Student122	0	#N/A	0	0

B.2 Fall 2018

Table B.3: Fall 2018 T-Test Data

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 001	3	6	7.5	6.5
Student 002	6.5	9.5	10	11
Student 003	7	11	10.5	9.5
Student 004	10.5	12	11	10
Student 005	7	6.5	8.5	7.5
Student 006	#N/A	10	#N/A	7.5
Student 007	5.5	10	12	11
Student 008	11	11	#N/A	11
Student 009	8	11	10	7.5
Student 010	9.5	12	11	11

Table B.3 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 011	9.5	12	11	12
Student 012	9.5	10.5	#N/A	10.5
Student 013	11	12	12	#N/A
Student 014	6.5	11.5	9.5	12
Student 015	5	10	10	10
Student 016	10.5	11.5	11.5	9.5
Student 017	9	7.5	11	11
Student 018	9	#N/A	11	12
Student 019	6	9	11	10
Student 020	5	7.5	8.5	6.5
Student 021	6	10	11	11
Student 022	7	8.5	9.5	9
Student 023	6	10	9	12
Student 024	#N/A	#N/A	#N/A	10
Student 025	9.5	12	#N/A	12
Student 026	6	8	8.5	9
Student 027	#N/A	9	11	11
Student 028	5.5	9	11	12
Student 029	#N/A	7	9	9
Student 030	#N/A	11	11	12
Student 031	7.5	8.5	8.5	10.5
Student 032	3.5	9.5	9.5	9.5
Student 033	10	11	12	#N/A

Table B.3 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 034	#N/A	#N/A	7.5	#N/A
Student 035	10	12	11	#N/A
Student 036	9.5	10.5	11	9
Student 037	6.5	8.5	8.5	7.5
Student 038	6	7.5	9	8.5
Student 039	10.5	12	10.5	12
Student 040	9	12	12	#N/A
Student 041	11	12	12	12
Student 042	#N/A	7	#N/A	#N/A
Student 043	7.5	9.5	7	8
Student 044	6.5	9.5	#N/A	9
Student 045	7	4.5	8.5	8.5
Student 046	6	6	7	7
Student 047	8.5	8.5	10.5	9
Student 048	7.5	9.5	10	8
Student 049	#N/A	3	2.5	4
Student 050	5.5	9	#N/A	5.5
Student 051	9.5	12	12	12
Student 052	7.5	11	9.5	11
Student 053	6	9	8	11
Student 054	7.5	9	11	11
Student 055	8.5	#N/A	10	10
Student 056	5	10	11	10

Table B.3 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 057	9.5	11	12	11
Student 058	7.5	11.5	10.5	8.5
Student 059	8.5	7	10	12
Student 060	6	7	9.5	9.5
Student 061	6	12	10.5	10.5
Student 062	5	7.5	8.5	7
Student 063	6	6.5	8.5	8
Student 064	11	12	#N/A	11
Student 065	10	7.5	11.5	12
Student 066	9	11	10	10
Student 067	3.5	9.5	6.5	8
Student 068	11.5	12	12	12
Student 069	11	12	12	12
Student 070	7	8	10.5	7
Student 071	4.5	5	7.5	9
Student 072	9.5	10	12	11
Student 073	5	10.5	10.5	9.5
Student 074	12	11	12	#N/A
Student 075	6	11	10	9.5
Student 076	5	7.5	10	#N/A
Student 077	11	12	11	12
Student 078	9	9	11	11
Student 079	10	11	10	12

Table B.3 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 080	7	10	11	10
Student 081	10.5	12	11	11
Student 082	2	6.5	6	5
Student 083	5	8	9.5	7
Student 084	9.5	11	11	11
Student 085	10	11	12	12
Student 086	10.5	9	#N/A	#N/A
Student 087	6	9	10	8
Student 088	9.5	12	11	12
Student 089	11	12	12	12
Student 090	9.5	12	12	11.5
Student 091	12	12	12	11
Student 092	11.5	12	11	10
Student 093	10	10.5	10.5	9.5
Student 094	8.5	11	11	8
Student 095	11	7.5	9	7
Student 096	10.5	10.5	9	9.5
Student 097	10	12	10.5	11
Student 098	6.5	10	8.5	7.5
Student 099	7.5	10.5	10.5	10.5
Student 100	10	12	11	11
Student 101	7.5	5	11	10
Student 102	5	7.5	10.5	11

Table B.3 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 103	6	10	11	11
Student 104	4.5	10	9	#N/A
Student 105	11	11	11	10
Student 106	8.5	12	10.5	11
Student 107	6.5	9.5	11	11
Student 108	11	11	11	10
Student 109	7.5	#N/A	8	8
Student 110	8	9	10.5	11
Student 111	8.5	#N/A	10	11
Student 112	9	10	9	9
Student 113	6	#N/A	#N/A	#N/A
Student 114	10.5	11	11	12
Student 115	8.5	10	10	9
Student 116	6.5	9.5	10	11
Student 117	7	10.5	9.5	11
Student 118	11	11	12	11
Student 119	5	9	9	8.5

Table B.4: Fall 2018 Question 10-12 T-Test Data

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 001	0	1	1.5	0.5

Table B.4 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 002	1.5	1.5	2	2
Student 003	1	2	1.5	1.5
Student 004	2.5	3	2	1
Student 005	1	0.5	0.5	0.5
Student 006	#N/A	2	#N/A	1.5
Student 007	1.5	2	3	2
Student 008	2	2	#N/A	2
Student 009	3	2	2	1.5
Student 010	2.5	3	3	3
Student 011	1.5	3	3	3
Student 012	2.5	2.5	#N/A	1.5
Student 013	3	3	3	#N/A
Student 014	1.5	2.5	1.5	3
Student 015	1	1	1	1
Student 016	2.5	2.5	2.5	1.5
Student 017	0	1.5	3	3
Student 018	2	#N/A	3	3
Student 019	1	1	2	1
Student 020	0	1.5	1.5	0.5
Student 021	1	2	2	2
Student 022	0	0.5	1.5	2
Student 023	1	1	1	3
Student 024	#N/A	#N/A	#N/A	1

Table B.4 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 025	2.5	3	#N/A	3
Student 026	1	1	1.5	1
Student 027	#N/A	1	2	2
Student 028	0.5	2	2	3
Student 029	#N/A	2	2	2
Student 030	#N/A	3	2	3
Student 031	1.5	1.5	1.5	1.5
Student 032	0.5	2.5	1.5	0.5
Student 033	3	3	3	#N/A
Student 034	#N/A	#N/A	0.5	#N/A
Student 035	2	3	2	#N/A
Student 036	1.5	1.5	2	1
Student 037	2.5	1.5	1.5	0.5
Student 038	1	1.5	2	1.5
Student 039	1.5	3	2.5	3
Student 040	3	3	3	#N/A
Student 041	3	3	3	3
Student 042	#N/A	2	#N/A	#N/A
Student 043	0.5	1.5	1	1
Student 044	1.5	0.5	#N/A	2
Student 045	2	0.5	0.5	0.5
Student 046	1	1	2	2
Student 047	2.5	1.5	2.5	1

Table B.4 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 048	1.5	1.5	1	1
Student 049	#N/A	0	0.5	1
Student 050	1.5	2	#N/A	1.5
Student 051	2.5	3	3	3
Student 052	2.5	2	2.5	3
Student 053	1	2	1	3
Student 054	0.5	2	3	2
Student 055	1.5	#N/A	2	2
Student 056	1	2	2	1
Student 057	2.5	3	3	3
Student 058	1.5	2.5	1.5	1.5
Student 059	2.5	1	1	3
Student 060	3	2	1.5	1.5
Student 061	0	3	1.5	1.5
Student 062	1	1.5	1.5	1
Student 063	1	1.5	0.5	2
Student 064	3	3	#N/A	2
Student 065	2	1.5	2.5	3
Student 066	2	2	2	1
Student 067	1.5	2.5	0.5	1
Student 068	2.5	3	3	3
Student 069	3	3	3	3
Student 070	0	1	2.5	0

Table B.4 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 071	0.5	1	0.5	2
Student 072	1.5	2	3	2
Student 073	0	2.5	1.5	0.5
Student 074	3	2	3	#N/A
Student 075	1	2	2	1.5
Student 076	1	1.5	2	#N/A
Student 077	2	3	3	3
Student 078	3	2	2	2
Student 079	2	3	2	3
Student 080	1	1	2	1
Student 081	1.5	3	2	2
Student 082	1	1.5	1	0
Student 083	1	1	0.5	0
Student 084	1.5	2	2	2
Student 085	3	2	3	3
Student 086	1.5	1	#N/A	#N/A
Student 087	2	2	2	2
Student 088	1.5	3	2	3
Student 089	3	3	3	3
Student 090	2.5	3	3	2.5
Student 091	3	3	3	2
Student 092	2.5	3	2	1
Student 093	2	2.5	1.5	1.5

Table B.4 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 094	1.5	3	2	2
Student 095	3	1.5	1	1
Student 096	2.5	1.5	0	0.5
Student 097	1	3	2.5	3
Student 098	0.5	2	0.5	0.5
Student 099	0.5	2.5	1.5	2.5
Student 100	2	3	2	2
Student 101	1.5	1	2	1
Student 102	0	1.5	2.5	2
Student 103	2	2	3	3
Student 104	0.5	1	1	#N/A
Student 105	2	2	2	2
Student 106	1.5	3	2.5	3
Student 107	0.5	1.5	2	3
Student 108	2	2	2	1
Student 109	2.5	#N/A	2	1
Student 110	2	1	1.5	2
Student 111	0.5	#N/A	1	2
Student 112	1	1	1	1
Student 113	2	#N/A	#N/A	#N/A
Student 114	2.5	3	3	3
Student 115	0.5	2	1	1
Student 116	1.5	2.5	2	2

Table B.4 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student 117	1	2.5	1.5	3
Student 118	2	3	3	3
Student 119	2	2	2	1.5

B.3 Spring 2019

Table B.5: Spring 2019 T-Test Data

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student001	8.5	8.5	11	#N/A
Student002	8	11	12	#N/A
Student003	1.5	8.5	5	6
Student004	4.5	9	12	#N/A
Student005	#N/A	#N/A	9.5	9.5
Student006	8	12	11	#N/A
Student007	#N/A	11	10	10
Student008	9	#N/A	10	11
Student009	8	10	11	10
Student010	7.5	10.5	11	10
Student011	11	10.5	10	#N/A
Student012	8.5	8.5	7.5	7.5

Table B.5 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student013	1	#N/A	8.5	4.5
Student014	#N/A	12	#N/A	#N/A
Student015	10.5	10	11	#N/A
Student016	5	#N/A	10	12
Student017	8	11	#N/A	12
Student018	3.5	9.5	7.5	6
Student019	3	8	4	8
Student020	5.5	5	5	8.5
Student021	3	8	#N/A	10
Student022	4.5	9	#N/A	9
Student023	#N/A	#N/A	10.5	#N/A
Student024	7.5	#N/A	11	10
Student025	11	12	12	11
Student026	#N/A	11.5	12	#N/A
Student027	5	8	9.5	9
Student028	4.5	10.5	8.5	9
Student029	9	10	11	12
Student030	12	11	11	11
Student031	#N/A	#N/A	#N/A	8
Student032	8	12	10	11
Student033	11	#N/A	11	#N/A
Student034	12	12	12	12
Student035	2	8.5	7.5	9

Table B.5 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student036	3	11	10	9
Student037	7	7.5	8.5	7.5
Student038	8	12	11	#N/A
Student039	6.5	10.5	8.5	#N/A
Student040	5	12	11	12
Student041	10.5	11	11	11
Student042	6	10.5	9.5	9.5
Student043	9	10	11	12
Student044	1.5	7.5	#N/A	#N/A
Student045	11	9	#N/A	10
Student046	4	5.5	#N/A	9.5
Student047	5.5	10	5.5	9.5
Student048	8.5	11.5	10.5	9.5
Student049	4.5	7	9	8
Student050	5.5	9	9	12
Student051	10.5	#N/A	10	10.5
Student052	11	11	10	10
Student053	6	#N/A	6.5	7.5
Student054	12	11	11	11.5
Student055	12	12	12	12
Student056	11	12	12	10
Student057	6	11	12	12
Student058	9	10	9	10

Table B.5 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student059	8.5	10	10	9
Student060	2	7.5	9.5	9.5
Student061	4.5	8.5	8	6.5
Student062	8	10	10	10
Student063	5.5	9	10	8
Student064	5	10.5	11.5	9.5
Student065	12	12	12	12
Student066	2.5	9	10.5	9.5
Student067	#N/A	#N/A	#N/A	9
Student068	7.5	10.5	9.5	10.5
Student069	2	9	10	#N/A
Student070	8	9	11	11
Student071	9.5	11	9	10
Student072	2	5	3	1
Student073	11	12	#N/A	#N/A
Student074	6	#N/A	9	10
Student075	4.5	6.5	10	7
Student076	2	#N/A	5	0
Student077	6	11	10	10
Student078	7	#N/A	#N/A	#N/A
Student079	5	#N/A	#N/A	6
Student080	5	6.5	9.5	9.5
Student081	5	7	8.5	9.5

Table B.5 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student082	1	6	6	3.5

Table B.6: Spring 2019 Question 10-12 T-Test Data

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student001	0.5	2.5	2	#N/A
Student002	2	3	3	#N/A
Student003	0.5	2.5	1	0
Student004	1.5	1	3	#N/A
Student005	#N/A	#N/A	1.5	1.5
Student006	1	3	3	#N/A
Student007	#N/A	3	2	2
Student008	1	#N/A	2	2
Student009	2	3	2	2
Student010	1.5	2.5	2	2
Student011	3	1.5	1	#N/A
Student012	1.5	0.5	0.5	0.5
Student013	1	#N/A	0.5	0.5
Student014	#N/A	3	#N/A	#N/A
Student015	2.5	2	2	#N/A
Student016	2	#N/A	3	3
Student017	0	2	#N/A	3

Table B.6 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student018	1.5	2.5	1.5	1
Student019	0	1	0	0
Student020	0.5	2	2	1.5
Student021	1	1	#N/A	2
Student022	0.5	2	#N/A	1
Student023	#N/A	#N/A	1.5	#N/A
Student024	0.5	#N/A	3	2
Student025	3	3	3	3
Student026	#N/A	2.5	3	#N/A
Student027	1	1	1.5	2
Student028	1.5	2.5	1.5	1
Student029	3	3	2	3
Student030	3	3	3	3
Student031	#N/A	#N/A	#N/A	1
Student032	2	3	2	2
Student033	3	#N/A	2	#N/A
Student034	3	3	3	3
Student035	0	2.5	1.5	2
Student036	0	2	2	1
Student037	1	1.5	1.5	1.5
Student038	1	3	2	#N/A
Student039	0.5	2.5	0.5	#N/A
Student040	1	3	2	3

Table B.6 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student041	2.5	2	2	2
Student042	0	1.5	0.5	1.5
Student043	2	2	2	3
Student044	0.5	2.5	#N/A	#N/A
Student045	3	1	#N/A	1
Student046	0	0.5	#N/A	1.5
Student047	0.5	3	0.5	1.5
Student048	1.5	2.5	1.5	0.5
Student049	1.5	2	2	1
Student050	0.5	1	1	3
Student051	2.5	#N/A	1	1.5
Student052	3	3	2	2
Student053	0	#N/A	1.5	1.5
Student054	3	3	3	2.5
Student055	3	3	3	3
Student056	3	3	3	2
Student057	2	3	3	3
Student058	1	2	1	1
Student059	1.5	1	1	1
Student060	1	2.5	2.5	1.5
Student061	1.5	1.5	1	0.5
Student062	1	2	1	2
Student063	0.5	1	1	1

Table B.6 – continued from previous page

ID	Pre-Test	Post-Test 1	Post-Test 2	Post-Test 3
Student064	2	2.5	2.5	1.5
Student065	3	3	3	3
Student066	0.5	3	1.5	1.5
Student067	#N/A	#N/A	#N/A	1
Student068	1.5	2.5	0.5	1.5
Student069	0	2	3	#N/A
Student070	2	2	2	2
Student071	2.5	2	1	1
Student072	0	1	0	0
Student073	3	3	#N/A	#N/A
Student074	0	#N/A	2	2
Student075	1.5	1.5	1	2
Student076	0	#N/A	0	0
Student077	1	2	2	2
Student078	1	#N/A	#N/A	#N/A
Student079	1	#N/A	#N/A	1
Student080	1	2.5	1.5	1.5
Student081	1	1	0.5	0.5
Student082	0	0	0	0.5