Teaching and Learning Programming Using Code Animation

Y. Daniel Liang
Department of Computer Science, Georgia Southern University, USA

Abstract - *Code animation is a visual tool for tracing the execution of the code. It is developed using HTML5, CSS, and JavaScript. It is platform independent. It can be viewed from a browser on any device. We have developed code animation for more than 240 code examples in Java, C++, and Python to assist instructors to teach programming and help students learn difficult programming concepts and techniques visually and interactively. This paper presents several code animation examples for teaching and learning Java.*

Keywords: Animation, teaching programming, computer science education

1 Introduction

Students learn programming through examples. The key programming concepts and techniques are best demonstrated through good examples. Many new students find that programming textbooks are dry and boring, because the examples in a print book are static, not engaging, and hard to follow. We developed interactive code animation for demonstrating key programming concepts on variables, data types, constants, expressions, control statements, methods, pass by value, arrays, multidimensional arrays, and classes, and objects. The interactive code animation not only engage students, but also is effective to make the difficult concept easy to learn. We have developed more than 240 code animations for Java, C++, and Python. The code animation have been integrated in the Pearson's interactive ebooks [1, 2, 3], which have received positive reviews [4, 5]. This paper presents the code animation for the Java language. Specifically, we use the following examples to demonstrate how the code animation can help students to learn variables, selection statements, loops, methods, pass by value, and arrays effectively:

- ComputeArea.java
  https://liveexample.pearsoncmg.com/codeanimation/ComputeArea.html
- ComputeAndInterpretBMI.java
  https://liveexample.pearsoncmg.com/codeanimation/ComputeAndInterpretBMI.html
- RepeatAdditionQuiz.java
  https://liveexample.pearsoncmg.com/codeanimation/RepeatAdditionQuiz.html
- Palindrome.java
  https://liveexample.pearsoncmg.com/codeanimation/Palindrome.html
- TestMax.java
  https://liveexample.pearsoncmg.com/codeanimation/TestMax.html
- Increment.java
  https://liveexample.pearsoncmg.com/codeanimation/Increment.html
- SelectionSort.java
  https://liveexample.pearsoncmg.com/codeanimation/SelectionSort.html
- TestArrayArguments.java
  https://liveexample.pearsoncmg.com/codeanimation/TestArrayArguments.html

2 Animation for Variables

A variable represents a value stored in the computer's memory. This is the first important concept for new students. Our code animation shows how to declare variables and how to assign values to variables. We demonstrate it using program ComputeArea.java (https://liveexample.pearsoncmg.com/codeanimation/ComputeArea.html), as shown in Figure 2a. The program computes the area of a circle given the radius. Clicking the Start Animation button to launch the animation. You will see a step-by-step execution of the code upon clicking the Next button. The code

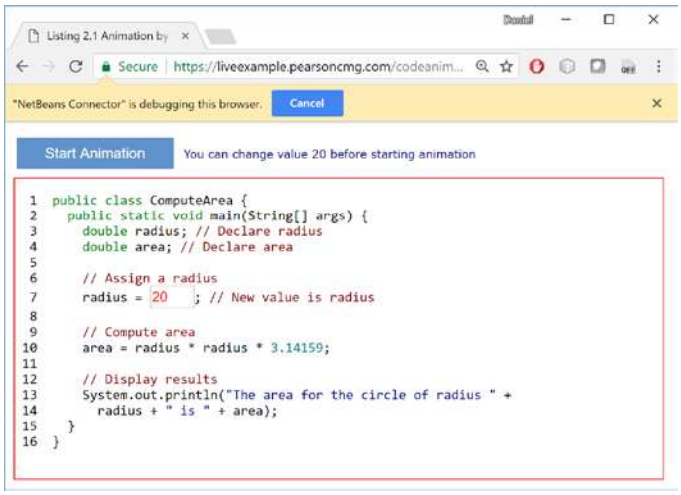

Figure 2a. Clicking the Start Animation button to launch animation.

currently being executed is highlighted in light green, as shown in Figure 2b. When executing line 3, the variable radius is declared. You will see the memory allocated for variable radius, but without value assigned. When executing line 7, a new value 20.0 is assigned to variable radius in Figure 2c. When the code in line 10 is executed, the expression radius * radius * 3.14149 is executed and its result is assigned to variable area, as shown in Figure 2d.

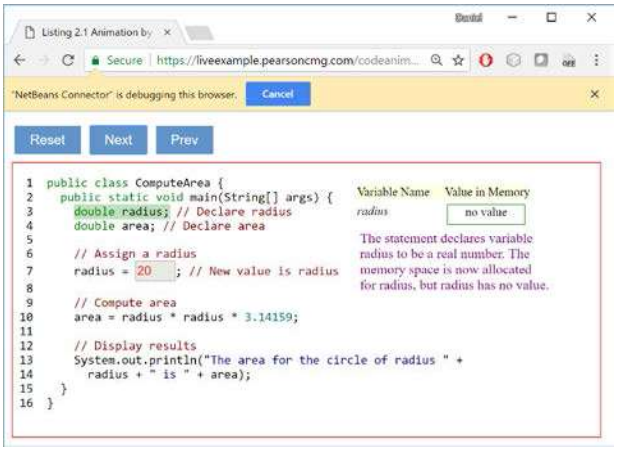

Figure 2b. Clicking the Next button to execute the next statement.

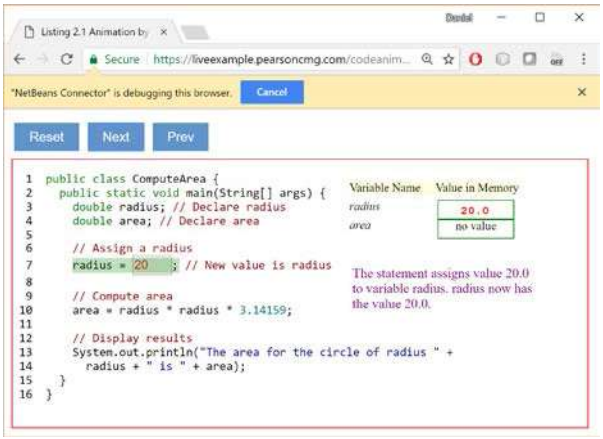

Figure 2c. Value 20 is assigned to variable radius.

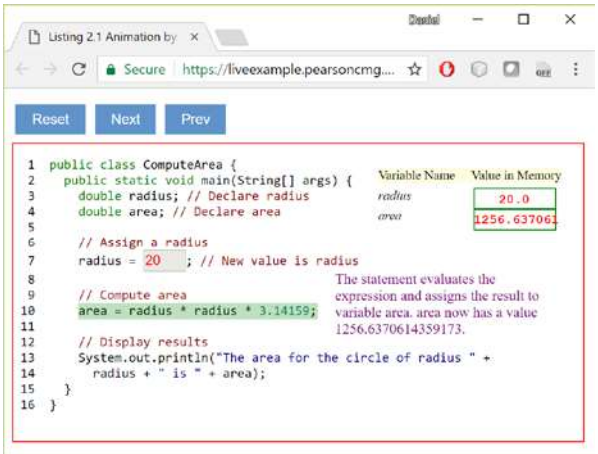

Figure 2d. A new value is assigned to area.

3 Animation for Selection Statements

Selection statements let you choose actions with alternative courses. A code animation can effectively demonstrate how a selection statement is executed. Let us look at the program ComputeAndInterpretBMI.java for computing and interpreting BMI (https://liveexample.pearsoncmg.com/codeanimation/ComputeAndInterpretBMI.html). The program prompts the user to enter the weight in pounds and height in inches and displays the BMI and interprets BMI. When the code in lines 8-9 are executed, the program prompts the user to enter the weight in pounds, as shown in Figure 3a. As you continue step through the code, you will encounter the if statement. The animation tests student understanding of the if statement by asking the question "Which line will be executed after this line?", as shown in Figure 3b. If the user enters the answer 28, the animation will respond that "You are correct."

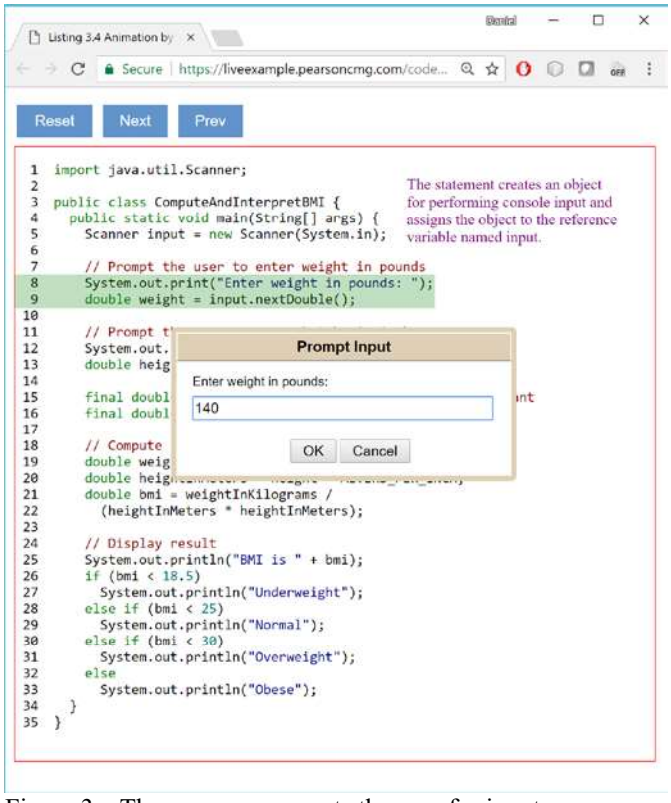

Figure 3a. The program prompts the user for input.

4 Animation for Repetition Statements

A loop can be used to execute statements repeatedly. Java has three types of loops: the while loop, the do-while loop, and the for loop. We give an example of using the while loop in this section. The program RepeatAdditionQuiz.java (https://liveexample.pearsoncmg.com/codeanimation/RepeatAdditionQuiz.html) randomly generates two single-digit integers and prompts the user for an answer for the addition of the two integers until the answer is correct. As you step

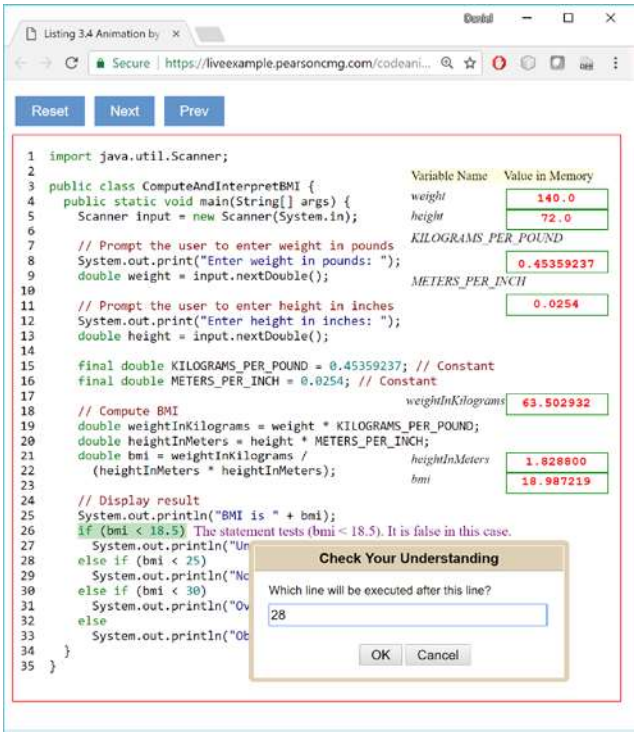

Figure 3b. The animation tests student understanding of the selection statement.

through the code, you will encounter the while statement. The animation tests student understanding of the while loop with a question "Which line will be executed after this line?", as shown in Figure 4a. If the user's answer is 16, the animation will display that "You are correct", as shown in Figure 4b. The animation continues to trace the execution of the while loop as you click the Next button.

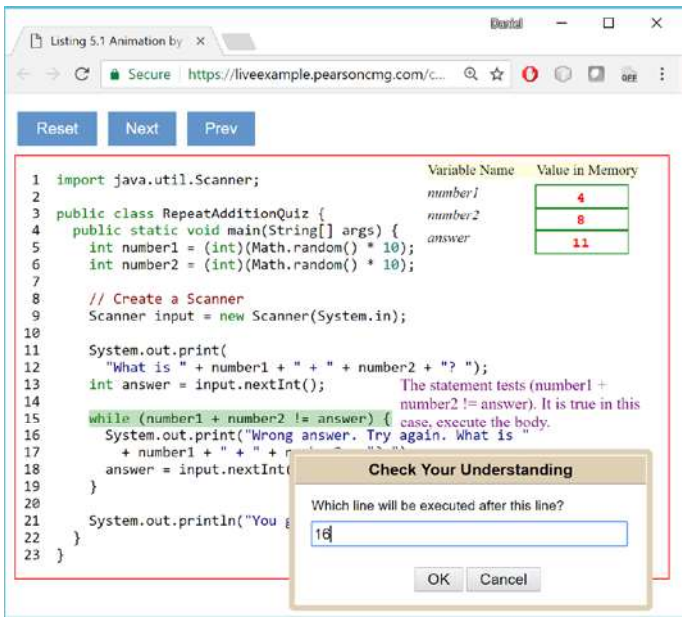

Figure 4a. The animation tests student understanding of the while loop.

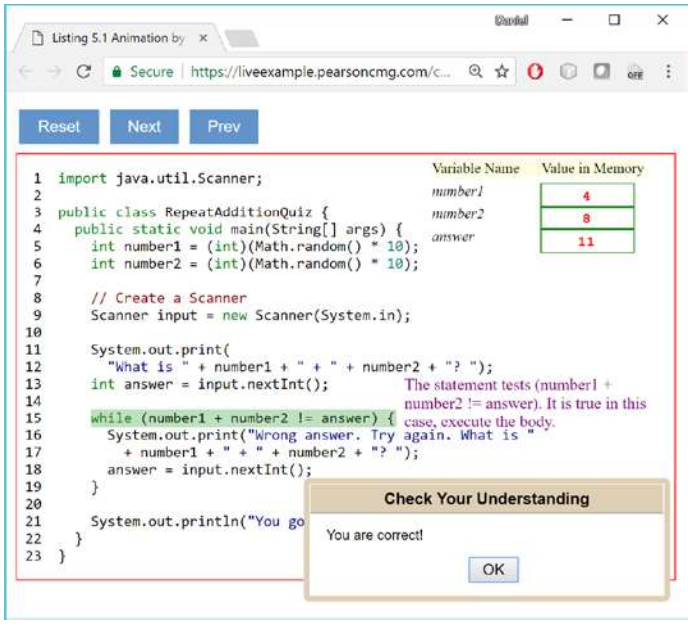

Figure 4b. The animation responds to student answer.

5 Animation for the break Statement

To see how the break statement can be used to simplify coding and how it works, let us look at the Palindrome program (https://liveexample.pearsoncmg.com/codeanimation/Palindrome.html). The program prompts the user to enter a string and

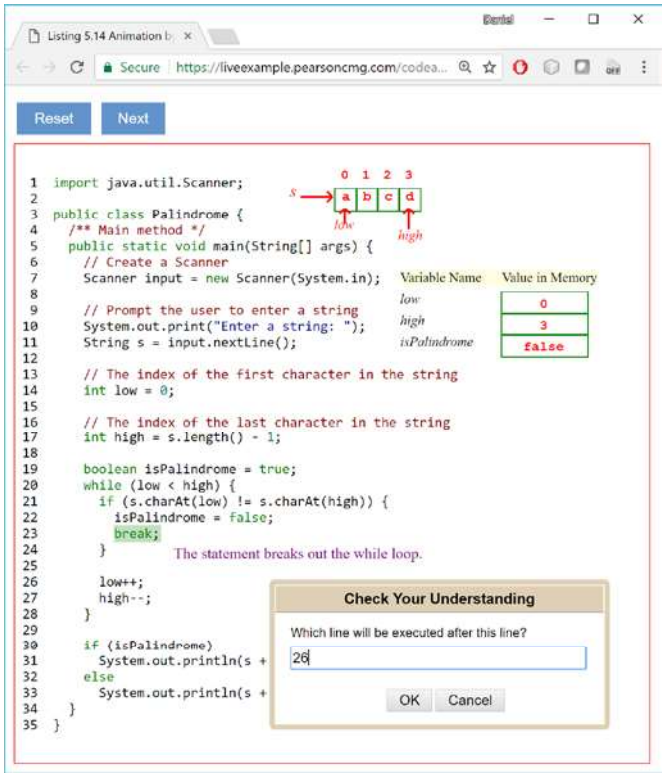

Figure 5a. The animation tests student understanding of the while loop.

tests if the string is a palindrome. As you step through the code, you will encounter the break statement in line 23. The animation tests student understanding of the break statement with a question "Which line will be executed after this line?", as shown in Figure 5a. If the user's answer is 26, the animation will display that "Wrong. The next line to execute is 30", as shown in Figure 5b.

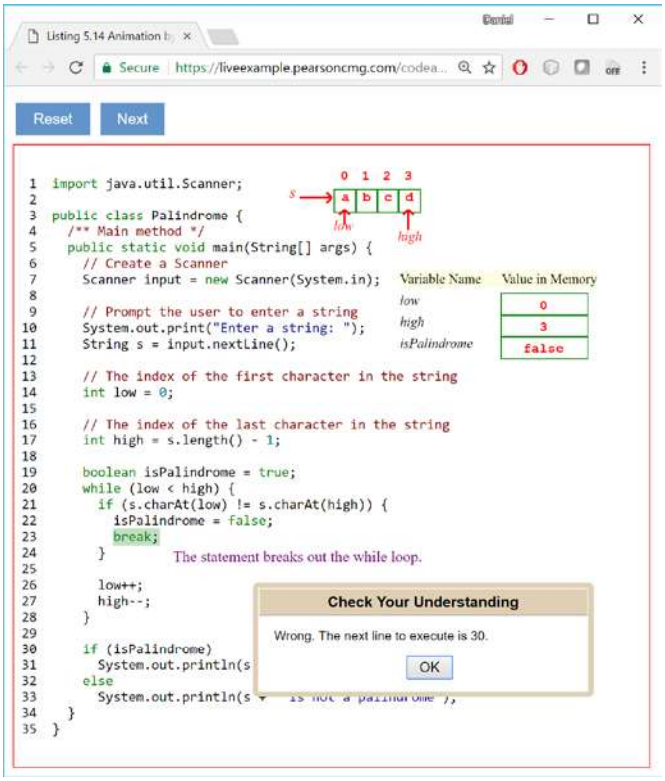

Figure 5b. The animation responds to student answer.

6 Animation for Methods

Defining methods and invoking methods is a challenge subject to teach. Students often find it difficult to understand how arguments are passed. An animation that shows how a method is invoked interactively can effectively convey the concept and help students see the big picture and as well as the small details on parameter passing. Let us look at the program TestMax.java (https://liveexample.pearsoncmg.com/codeanimation/TestMax.html). The program starts the execution from the main method and it then invokes the max method in line 6, as shown Figure 6a. When the max is called, the values of i and j are passed to num1 and num2, as shown in Figure 6b. When the return statement is executed, the method is finished and it returns a value back to its caller, as shown in Figure 6c. The returned value in the caller is assigned to variable k, as shown in Figure 6d.

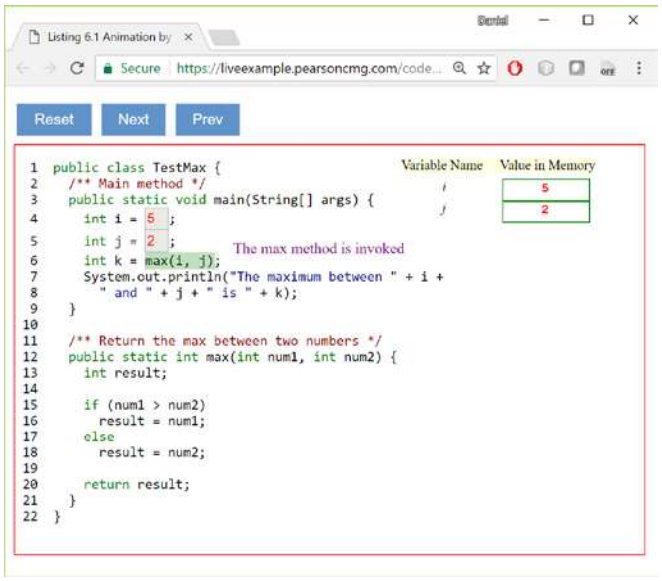

Figure 6a. The max method is to be invoked.

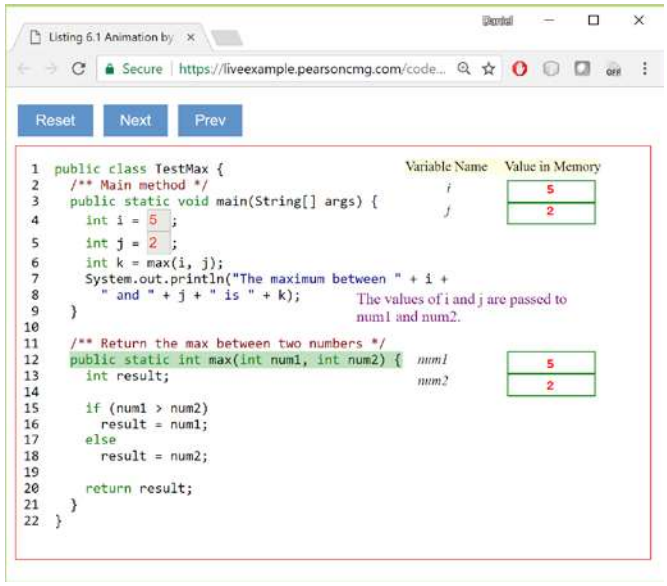

Figure 6b. When invoking max(i, j), the values of i and j are passed to num1 and num2.

7 Animation for Pass By Value

Java passes arguments by value. To help student grasp this concept, we use the Increment.java program (https://liveexample.pearsoncmg.com/codeanimation/Increment.html), as shown in Figure 7a. The program invokes method increment(x). The value of x is passed to n, as shown in Figure 7b. The method increases n by 1. Variable n becomes 2 now, as shown in Figure 7c. After the method is finished, the local variable n is removed from the memory and the control is back to the caller. The variable x remains 1, as shown in Figure 7d.

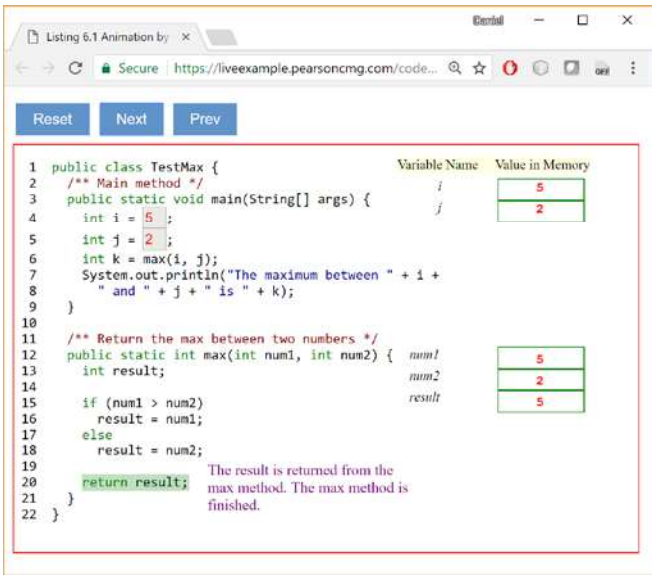

Figure 6c. The result is returned from the max method.

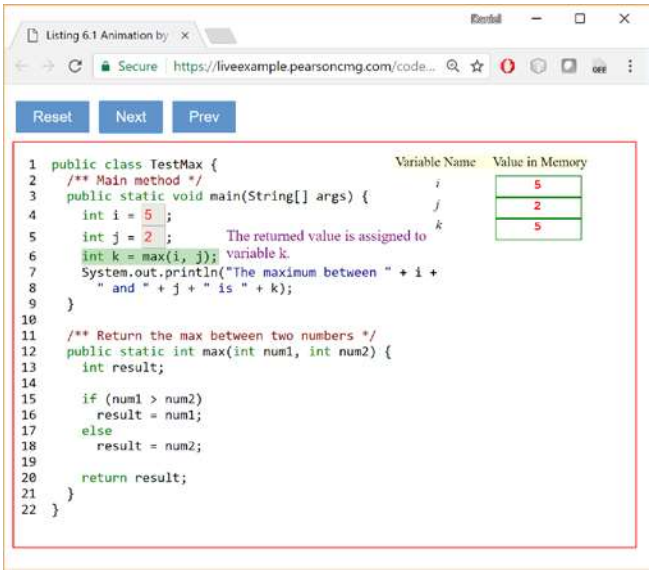

Figure 6d. The result is returned and the returned value is assigned to k.

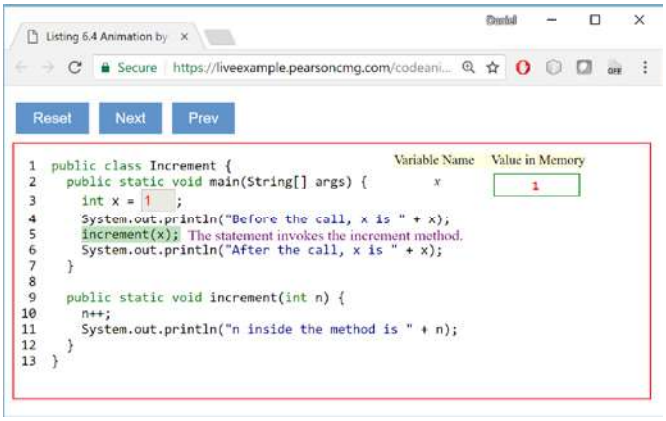

Figure 7a. The variable x is initialized to 1.

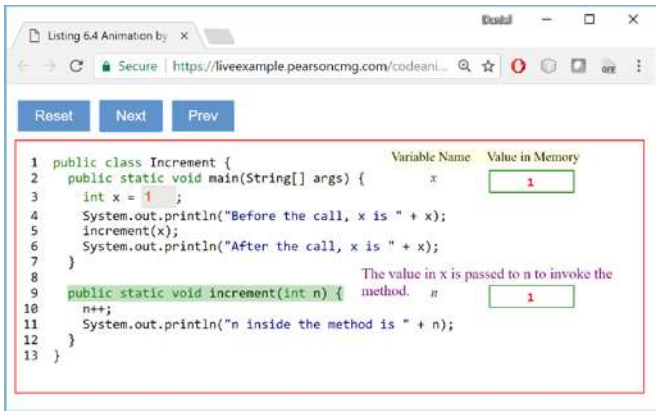

Figure 7b. The value of x is passed to n array is created with initial values.

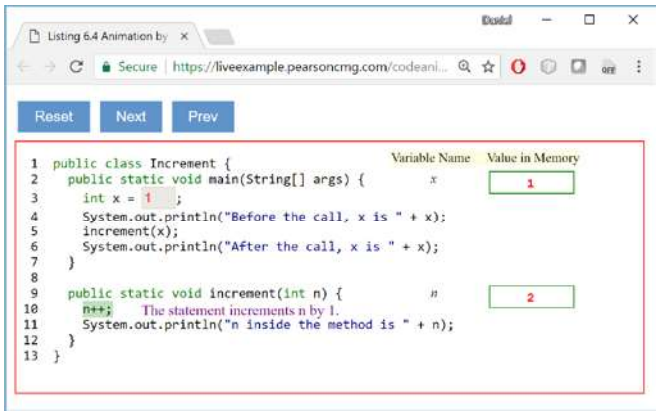

Figure 7c. The variable n is increased by 1.

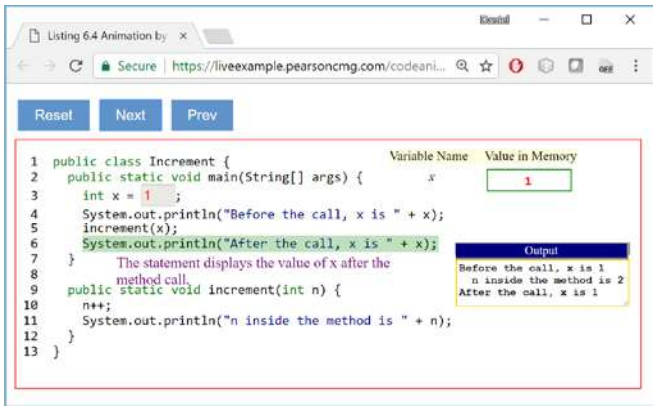

Figure 7d. After the method increment(x) is finished, local variable n is gone.

8 Animation for Arrays

An array represents a sequence of values. It is a very useful data structure for storing and processing a collection of data. Code animation can be used to demonstrate how data is stored in an array and how to access the data in the array through array index variable. Let us look at the SelectionSort.java program (https://liveexample.pearsoncmg.com/codeanimation/SelectionSort.html). The program creates an array with initial values, as shown in Figure 8a. It invokes the selectionSort method to sort

the array. The parameter list in the selectionSort method is now pointing to the array, as shown in Figure 8b. The selectionSort method finds the smallest value in the array and swaps it with the first value in the array, as shown in Figure 8c. The method continues the same process of finding the smallest value and swaps it with the first one in the remaining subarray, as shown in Figure 8d.

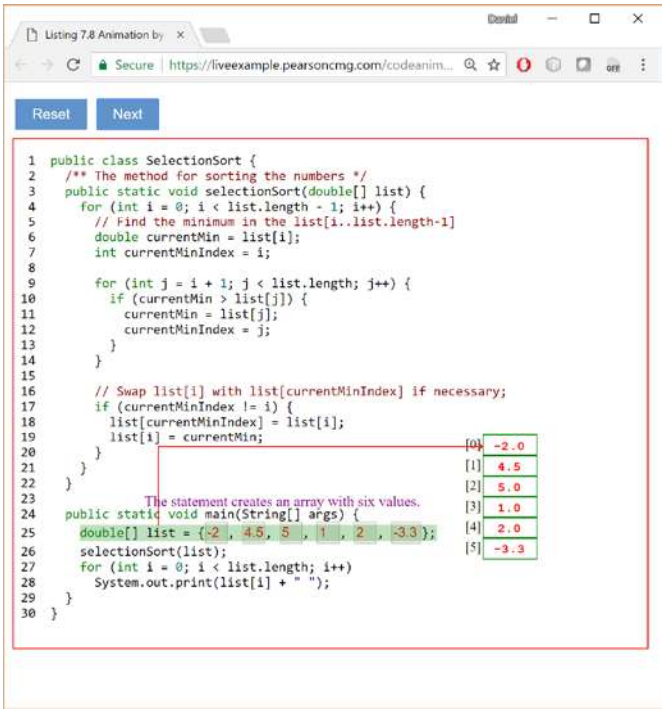

Figure 8a. The array is created with initial values.

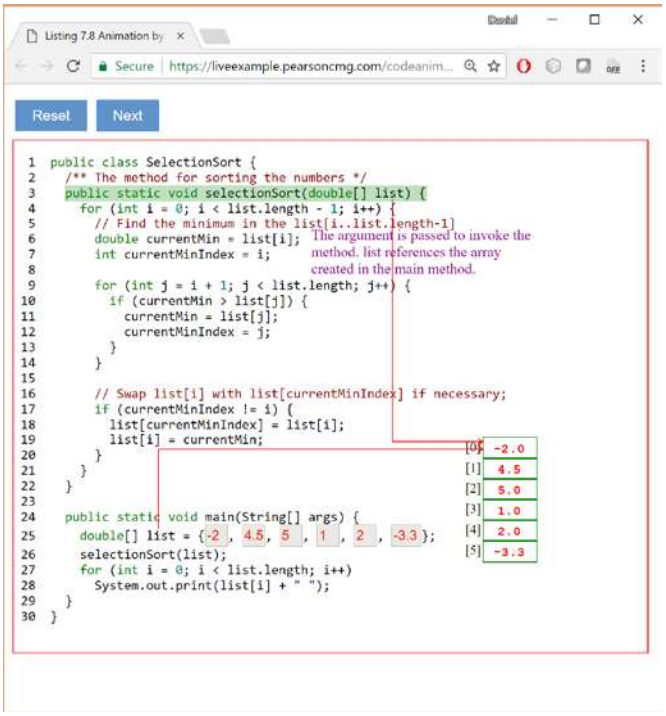

Figure 8b. The parameter list in the selectionSort method points to the array.

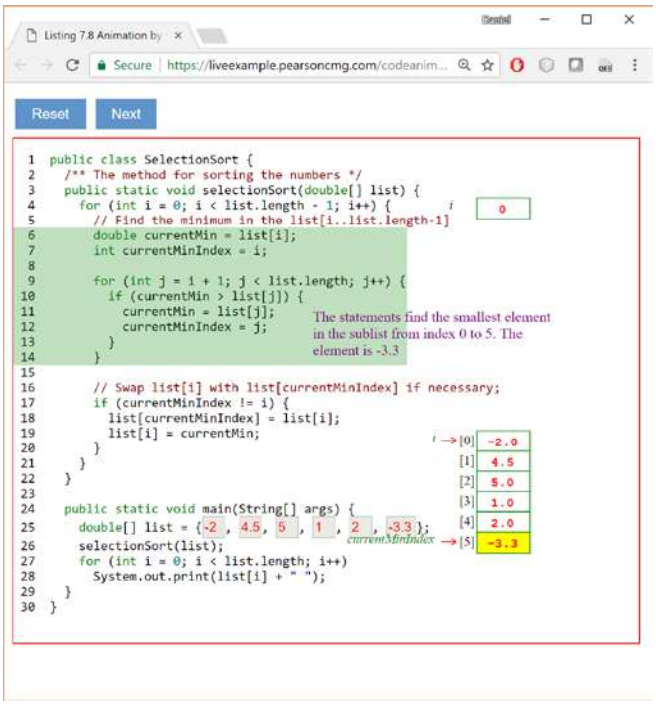

Figure 8c. The smallest value -3.3 will be swapped with the first value -2.0.

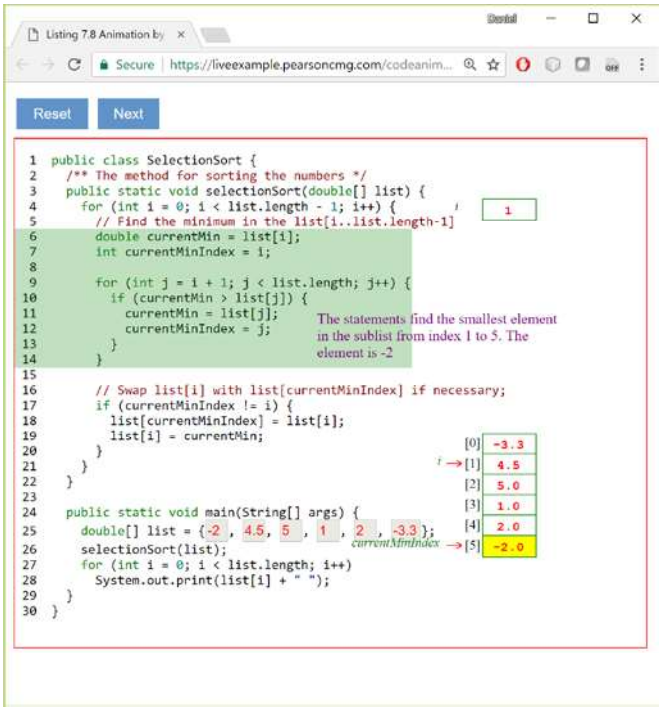

Figure 8d. The smallest value -2.0 and the fist value 4.5 in the remaining subarray will be swapped.

9 Animation for Passing Arrays to Methods

Java passes arguments by value. For an argument of a primitive type, the argument's value is passed. For an

argument of an array type, the value of the argument is a reference to an array; this reference value is passed to the method. Semantically, it can be best described as *pass-by-sharing*, that is, the array in the method is the same as the array being passed. Thus, if you change the array in the method, you will see the change outside the method. This is often a difficult concept for students. To help student grasp this concept, we use the TestArrayArguments.java program (https://liveexample.pearsoncmg.com/codeanimation/TestArrayArguments.html), as shown in Figure 9a. The program invokes method m(x, y). The value of x is passed to number and the value of y is passed to numbers, as shown in Figure 9b. Note that since y contains the reference value for the array, now numbers and y point to the same array. The method assigns 1000 to number in line 13 and assigns 5555 to numbers[0], as shown in Figure 9c. After the method is finished, the local variables number and numbers are gone. However, the array is still there. It is referenced by y. y[0] is now 5555.

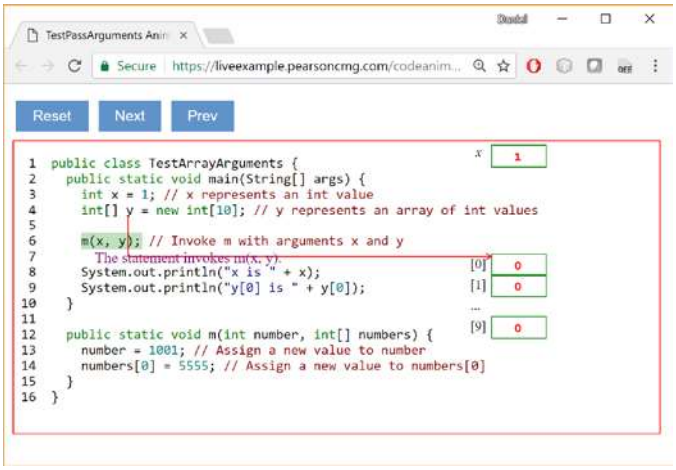

Figure 9a. The program invokes method m(x, y).

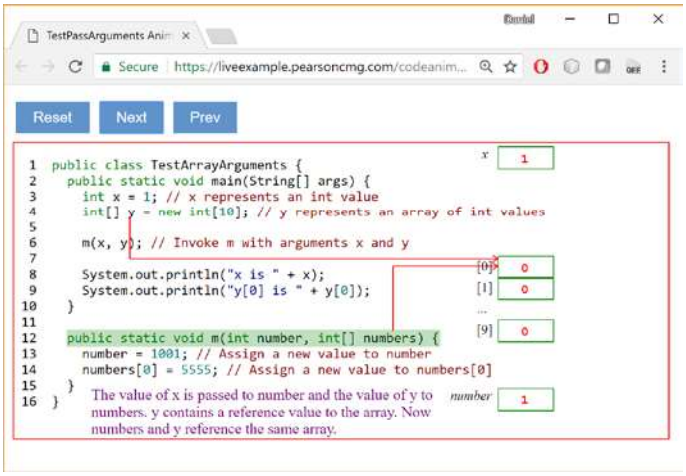

Figure 9b. The value of x is passed to number and the value of y is passed to numbers.

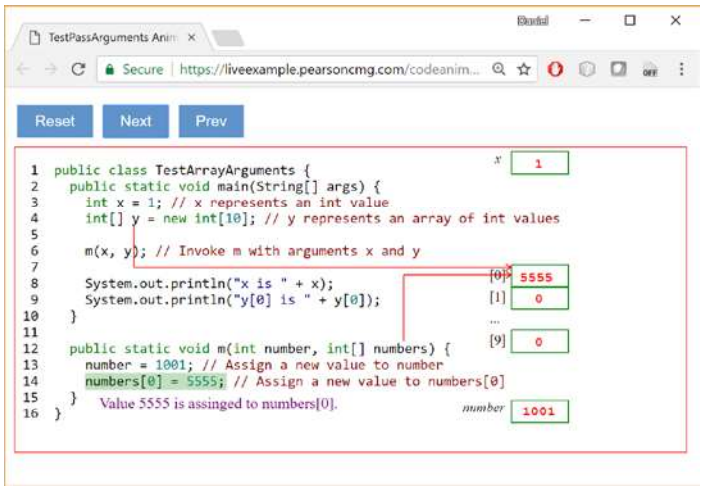

Figure 9c. 1001 is assinged to number and 5555 is assigned numbers[0].

10 Conclusions

Code animation is a great tool for instructors and students to teach and learn programming. It enables students to see the code execution visually and interactively. It helps novice students to grasp difficult concepts on variables, selection statements, loops, methods, pass by value, and arrays. We have used code animation in our interactive REVEL ebooks published by Pearson. Initially, we had one code animation per chapter. Instructors and students love it and provide us with a very positive feedback [4, 5]. Now the code animation is expanded, which includes almost all the examples in the first seven chapters to help new students get on track in programming. Our future work is to improve the code animation and create more effective code animations.

11 References

[1] Y. Daniel Liang, REVEL™ for Introduction Java Programming and Data Structures. ISBN-13: 978-0134167008. Pearson Education, 2016.

[2] Y. Daniel Liang, REVEL™ for Introduction to C++ Programming and Data Structures. ISBN-13: 978-0134669854. Pearson Education, 2018.

[3] Y. Daniel Liang, REVEL™ for Introduction to Python Programming and Data Structures. ISBN-13: 978-0135187753. Pearson Education, 2018.

[4] REVEL™ educator study observes homework and exam grades at University of Louisiana, Spring 2016, http://www.pearsoned.com/results/revel-educator-study-observes-homework-exam-grades-university-louisiana/

[5] REVEL educator study assesses quiz, exam, and final course grades at Central Michigan University, Fall 2015, http://www.pearsoned.com/results/revel-educator-study-assesses-quiz-exam-final-course-grades-central-michigan-university/