# Teacher’s guide

# Quadratic Sorting Algorithms

**OBJECTIVES:** The student will trace the execution of each of the following quadratic sorting algorithms: bubble, selection, and insertion sort.

The student will determine the approximate number of steps a sorting algorithm executes when sorting n items of data.

**ACTIVITIES/TIME:** Two Weeks

**MATERIALS:** Student Lesson A17: *Quadratic Sorting Algorithms*

Lab Assignment A17.1, *QuadSort*

Lab Assignment A17.1, Starter Code, *SortsTemplate.java, SortStep.java*

Worksheet A17.1*, Bubble Sort*

Worksheet A17.2*, Selection Sort*

Worksheet A17.3*, Insertion Sort*

Teacher’s Guide A17: *Quadratic Sorting Algorithms*Lesson A17 Handout, *Sorting Answers*

Lab Assignment A17.1 – Answers, *QuadraticSorts.java,* *QuadraticSortsCounting.java*

Worksheet A17.1, *Answer Sheet*

Worksheet A17.2, *Answer Sheet*

Worksheet A17.3, *Answer Sheet*

**REFERENCES:** **The Complete Collection of Algorithm Animations**  
<http://www.cs.hope.edu/~alganim/ccaa/sorting.html>  
This site provides a wide variety of animated sorting simulations.

**Interactive Data Structure Visualizations**  
<http://www.student.seas.gwu.edu/~idsv/idsv.html>

Another site with interactive data structure visualizations for quadratic sorts - animations for bubble, insertion and selection sorts.  
  
**The Object Oriented Programming Web: Animated Algorithms**

<http://oopweb.com/Algorithms/Documents/AnimatedAlgorithms/VolumeFrames.html>

More animated algorithms, including an insertion sort.

**INSTRUCTOR**

**NOTES:** Students are expected to have a working knowledge of sorting algorithms but are not required to recite code for the AP exam. In general, they need to know how a sorting algorithm works and the relative efficiency of each one. The student lesson presents three quadratic sorting algorithms: bubble sort, selection sort, and insertion sort.

To emphasize the differences between them, have the students role play the sorts. Ask the students to stand in a line at the front of the room. Show each of the methods of sorting using age, height, or name. Lots of repetition helps. After practicing, have one of the students do the sorting. The other students can help if an error occurs. This exercise can be scheduled at the beginning of class for a few days or every other day. Students will soon be able to distinguish between the sorts.

The category of quadratic sorts can be seen when students time any of these algorithms. As the size of the data set doubles, the time required increases as a quadratic function. Lab Assignment A17.1, *QuadSort* requires students to code all three sorting algorithms into a stubbed out program, *SortsTemplate.java*. The students count the number of steps for each sorting algorithm. Students preparing for the AB exam need to have a full understanding of order of algorithms as presented in Lesson AB25, *Order of Algorithms*.

Embedded in the student outline are small sets of 6 unsorted integers for practice, to evaluate student understanding of each sorting algorithm. (Answers for each sorting algorithm have been included on a two-page document, Lesson A17 Handout, *Sorting Answers*.) The student lesson is geared around sorting 6 integers. To explain the logic of each sorting algorithm, work with small lists of 6 integers on the board.

A shell program (*SortStep.java)* is provided for Lab Assignment A17.1, *QuadSort*. Students will simply need to replace the method stubs with the code covered in the student lesson. The sample solution for Lab Assignment A17.1 includes empirical data which could vary from student answers for two reasons: different analysis of the step requirements of each algorithm and the nature of the random numbers generated.

Students may encounter a warning in Lab Assignment A17.1 that can safely be ignored. The warning says that *Sorts.java* uses unchecked or unsafe operations. This is due to the fact that Comparable objects are not always comparable to each other (Apples and PassengerJets, for example). The Java compiler cannot tell if there will be a problem during run time. The method of fixing this warning is beyond the scope of this curriculum.

**WORKSHEET**

**NOTES:** Three worksheets give students practice working with the *Bubble Sort* (Worksheet A17.1), *Selection Sort* (Worksheet A17.2) and *Insertion Sort* (Worksheet A17.3) algorithms. Similar to the suggested practice in the student lesson for each algorithm, students are asked to re-write the contents of the array after each successive pass through the outer loop. This activity helps students understand the differences between the different sorting methods. Students are then asked to revise the algorithm to sort words (using the String class). The final problem exercise of each worksheet asks students to re-write the contents of the String array after each successive pass through the outer loop. Note that once the words are sorted, they form a readable (perhaps goofy) sentence.