**Problem**

By day 7 of this week, post your solution to the following problem in the assignments folder.  Post whatever you have completed, even if it is not working.

Sorting is a frequent array operation.  An array can be sorted if there is an ordering relation between the elements in the array.  For example, if the array consists of a set of integer elements, we can order the elements according to the integer number line.  A common method whereby the elements in an array may be sorted is called the bubble sort.  Assuming an array of positive integers which we wish to sort according to the sequence represented by the integer number line, the bubble sort operates as follows:

1.  Find the two adjacent elements, X and Y, in the array such that X > Y and swap X and Y, then sort the resulting array.

2.  If there is no adjacent pair of elements, X and Y, in the array such that X > Y , the list is sorted.

Note that the purpose of swapping two elements X and Y that occur out of order is so that after the swap, the new list is closer to a sorted list.  After a sufficient amount of swapping, we should end up with all the elements in order.  For example given the array:

**{1 2 5 4 7 3 6 8 10 9}**

we would commence by finding the elements 5 and 4 and swap them to get:

**{1 2 4 5 7 3 6 8 10 9}**

We would then continue as follows:

**{1 2 4 5 7 3 6 8 10 9}** **{1 2 4 5 3 7 6 8 10 9}** **{1 2 4 3 5 7 6 8 10 9}** **{1 2 3 4 5 7 6 8 10 9}** **{1 2 3 4 5 6 7 8 10 9}** **{1 2 3 4 5 6 7 8 9 10}**

The process is known as a bubble sort because elements slowly bubble up to their correct location.

**Design and implement a Java program which sorts a 10 element integer array using the bubble sort process.  The** **elements of the array to be sorted should be supplied by the user (assume the user will not input duplicates).**

Create a GUI front end for your bubble sort program.  The result should look something like that presented in **Figure 1**.  You may use any graphic element to create your display - JOptionPanes are the easiest and most basic tools available to you, but you may also experiment with additional GUI controls such as JFrames, JPanels, etc.  I would advise you to use objects from the Swing library (those objects begin with 'J') as opposed to objects from the AWT library - Swing objects are a little easier to use and are ultimately more flexible and robust than AWT objects.

In the example below, there are 10 text fields on a JFrame background to allow input of array elements (remember that for each text field you must press the carriage return key to invoke the listener).  When the array has been populated, we press the start button, at which point the given array is output as a label.  We then sort the result and output the sorted array.

**Hint:  Do not attempt to make the GUI too sophisticated.  Just concentrate on producing a working result (although sound programming techniques should still be applied).**



**Figure 1:** Bubble sort GUI (partial example)

Remember to write the source code for each class in a separate file which must have the same name as the class name together with the extension **.java**.  Remember also that by convention, class names commence with a capital letter.

As with all programs you write, you should provide a well-structured solution that is easy to read.  You should use meaningful identifier names and should provide useful comments.  A large proportion of the marks for this assignment will be based on the structure of your classes, not whether they do or don't work correctly.

**The focus of the assignment is for you to familiarise yourself with the Java concepts introduced in this seminar (arrays and introductory graphic elements).** If you have difficulty with the assignment, you may discuss these problems in the discussion folder as long as you do not directly discuss the assignment problems.

**In the event of compilation problems (i.e. your program will not compile), do not post entire classes and ask what is wrong with them.**  This would violate the above stricture of not discussing the assignment problems in the main folder.  Instead, try and identify the source of the problem through a process of elimination (by commenting out chunks of the code).  You may discuss error messages you may receive, you may discuss compilation problems with the applications found in the lecture, and perhaps these discussions will help you to figure out where your errors lie.  In addition, you may always contact me via private e-mail if you are having a specific problem and none of the above solutions seem to work for you.  Please use this option as a last resort, however!

It is a good idea to work in a step-by-step top-down manner.  For example, first define the general structure of a class with empty methods and compile the file.  Once it compiles successfully, then start adding further detail, recompiling after every few lines.  This way, the risk of getting a screen full of error messages is reduced, and you can be assured of handing in something that works by the end of the week.

Use the Safe Assign link below to submit your assignment.