

## CIT 149: Java I

### Chapter 7 Lab 3 Directions

In this lab we will write a program that will bubble sort an array. This program will consist of two separate classes. When the Demo program is run it will display as:



```
C:\Windows\system32\cmd.exe
Please enter number 1
23
Please enter number 2
1
Please enter number 3
13
Please enter number 4
106
Please enter number 5
2
Please enter number 6
99
Please enter number 7
15
Array values before sorting:
23 1 13 106 2 99 15
Array values after sorting:
1 2 13 15 23 99 106
Press any key to continue . . .
```

Here we request the user enter 7 numbers. The 7 numbers are added to an array and then displayed in the order entered. Afterwards the numbers are sorted and displayed in their sorted order.

Let's get started.

1. Open a new document window in TextPad and save the program as BubbleSorter.java.
2. Type the appropriate block comments, and the class header, and opening brace. This class is a helper class and will not contain the main method.

3. Our first method is the sort() method. As indicated, this method will contain a flag used for sorting. It will invoke the interchange() method to do the actual exchange when necessary. Type:

```
public static void sort(int[] a)
{
    boolean sorted = false;
    int iterations = a.length - 1; // Max number of passes

    while(!sorted && iterations > 0)
    {
        sorted = true;

        for (int i = 0; i < iterations; ++i)
        {
            if(a[i] > a[i + 1])
            {
                // Swap: bubble largest value up
                // to highest position

                interchange(i, a);
                sorted = false;
            }
        }
        --iterations;
    }
}
```

- First the sort() method has an argument of an array of integers named a. The value for this array will be received from the driver class.
- A boolean is set to false. This boolean will be used to determine whether a swap needs to be made.

- An integer named *iterations* is set to the length of the array minus 1. The reason for the minus 1 is that you will start with the second number when sorting. You never start with the first array element in the series.
- A while loop will run its code if the boolean IS NOT true AND the integers *iterations* is greater than zero.
- The boolean is used as a flag and is set to true.
- A for loop has a starting point of 0 and an ending point of when *iterations* is less than *i*. *i* will increase by 1 prior to the loop being run since the ++ is before *i* in the increment count.
- Within the for loop we have an if statement. The code for the if statement will be run if the array with an element of *i* is greater than a with an element of *i + 1*. If this statement results as true then:
  - the interchange() method is invoked and passed the values of both *i* and the array.
  - and the boolean *sorted* is set to false.
- After the for loop has completed the *iterations* variable is decreased by 1.

4. The interchange() method does the actual swapping. Type:

```
private static void interchange(int i, int[] array)
{
    int temp;
    temp = array[i];
    array[i] = array[i + 1];
    array[i + 1] = temp; // Original value of a[i]
}
```

- This method is passed the values of the integer *i* from the for loop in the sort() method and the value of the array.
- An integer named *temp* is declared. This integer will represent a temporary value for an array element.

- The array element with an index number of  $i$  is set to equal the value of the array element with an index number of  $i + 1$ .
  - Then `array[i + 1]` receives the original value of `a[i]` by setting it to equal the temp variable.
5. Close the class and compile the program. Fix any errors you may have.
  6. Our next step is to create the driver class to test the BubbleSort class.
  7. Open a new document in Textpad and save the program as BubbleSortDemo.java.
  8. Type the block comments as done previously, and also the import statement that will import the Scanner class in the java.util package.
  9. Type the class header, opening brace, main method header, and opening brace for the main method.
  10. First we will construct a Scanner object and an array with 7 elements. Type:

```
Scanner keyboard = new Scanner(System.in);
int[] b = new int[7];
```

- Arrays are not expandable. This means that the array can have only 7 integers
11. Using a for loop we will ask the user to enter the 7 numbers and add the numbers to the array. Type:

```
for(int i = 0; i < 7; i++)
{
    System.out.println("Please enter number " + (i+1));
    b[i]= keyboard.nextInt();
}
```

12. We will display a message that the first print out is the array before it is sorted. Type:

```
System.out.println("Array values before sorting:");
```

13. Declare an integer named *i*, which will be used in two separate for loops. Type:

```
int i;
```

14. A for loop will go through the array, printing out each value. Type:

```
for (i = 0; i < b.length; i++)  
    System.out.print(b[i] + " ");
```

- This is fairly simple to understand. It simply goes through the loop and prints out the value of the element with the current value of *i*.

15. We print out a blank line to separate the before and after sorting. Type:

```
System.out.println();
```

16. Next we want to sort. This is done by invoking the `sort()` method in the `BubbleSorter` class. Type:

```
BubbleSorter.sort(b);
```

17. We display a message that the next display will be the value of the array after sorting. Type:

```
System.out.println("Array values after sorting:");
```

18. A for loop is used to go through and print out the new array, after the sort() method sorted it. Type:

```
for (i = 0; i < b.length; i++)  
    System.out.print(b[i] + " ");
```

19. Finally a blank line is printed, simply for formatting purposes and we close the main method and class. Type:

```
        System.out.println();  
    }  
}
```

20. Compile the program and fix any errors if necessary.
21. Compress the following files into a single zip or rar file and submit to the appropriate drop box.

```
BubbleSorter.java  
BubbleSorter.class  
BubbleSortDemo.java  
BubbleSortDemo.class
```