Bubble Sort Exercise

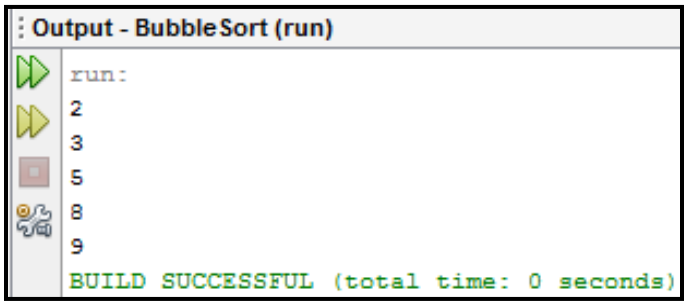
The bubble sort is the simplest sorting algorithm. We are going to sort a list of numbers in an array and print out the sorted array.

1. The name of this project and class should be **BubbleSort**.

```
/* Bubble sort is the simplest sorting algorithm
 * Written by
 * Date
 * JDK Version
 */

public class BubbleSort {

    public static void main(String[] args) {

        int number[] = {8, 5, 3, 2, 9};
        boolean swap = true;
        int temp;

        while (swap == true) {
            swap = false;
            for (int i = 0; i < number.length - 1; i++) {
                if (number[i] > number[i + 1]) {
                    temp = number[i + 1];
                    number[i + 1] = number[i];
                    number[i] = temp;
                    swap = true;
                }
            }
        }
        for (int i = 0; i < number.length; i++) {
            System.out.println(number[i]);
        }
    }
}
```

2. Type in the above program. Compile your program. Correct any typos that you made.

3. Execute this program. What did the program do? The numbers in the array (8, 5, 3, 2, 9) should have been sorted and printed as follows:



Explanation:

First Pass:
(**8 5** 3 2 9) ⟶ (**5 8** 2 3 9) Algorithm compares the first two elements and swaps them since 8>5
(5 **8 2** 3 9) ⟶ (5 **2 8** 3 9) Swap since 8 > 2
(5 2 **8 3** 9) ⟶ (5 2 **3 8** 9) Swap since 8>3
(5 2 3 **8 9**) ⟶ (5 2 3 **8 9**) No swap since 8 is not > 9

Second Pass:
(**5 2** 3 8 9) ⟶ (**2 5** 3 8 9) Swap since 5>2
(2 **5 3** 8 9) ⟶ (2 **3 5** 8 9) Swap since 5>3
(2 3 **5 8** 9) ⟶ (2 3 **5 8** 9) No swap since 5 is not > 8
(2 3 5 **8 9**) ⟶ (2 3 5 **8 9**) No swap since 8 is not > 9

Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.

Third Pass:
(**2 3** 5 8 9) ⟶ (**2 3** 5 8 9)
(2 **3 5** 8 9) ⟶ (2 **3 5** 8 9)
(2 3 **5 8** 9) ⟶ (2 3 **5 8** 9)
(2 3 5 **8 9**) ⟶ (2 3 5 **8 9**)