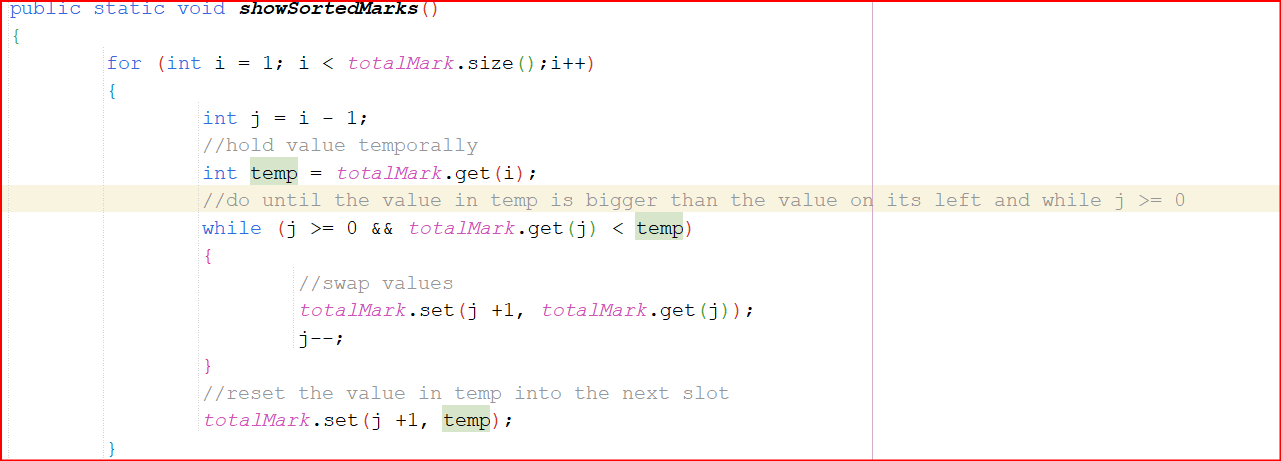
Insertion sort algorithm:

We start by identifying an array and fill it with data, then we start our process of arranging that data by using the insertion algorithm.

To do that the first step is to consider that the first element in that array is sorted, so we start from the second element in the array by using a for loop to iterate from the second element till the last element in the array, we use this syntax to do so.



The for loop starts as I mentioned from the second element and ends at the last element in the array to go through all the elements in the array.

Then we use another variable to save the index before the current index that we stand on or the one on its left, after that we use a variable to store the value of the field in the array that we stand at so we can insert into it and not lose the data that is stored in that field.

Then we start a while loop that keep the iterating until we see the correct slot to insert that data into, we do this by checking the j variable to see that it didn’t reach zero(checking that it is not zero means that it is not on the start of the array) and that the value that we are standing at in the array with the j variable is smaller than the data that we stored in the temp variable.

After checking that, we swap the value that is in the j index of the array to the next index and overwrite it with that data, but don’t forget that we saved the data for the j + 1 index in the temp variable, j + 1 is equal to the I variable, then we decrement the j variable by one.

We keep doing this until one of the conditions are not met, then we break out of the while loop(if j becomes less than zero or the array[j] value is bigger than the value in the temp variable), after breaking we insert the temp variable into the correct array position which is in this case j + 1, then we repeat this process until we go over all the elements in the array.

Note that it may get different based on if you want to sort the array in ascending order or descending order(the condition within the while loop will be different from bigger than the temp variable to smaller than it depending on the case).

Here is an example of an sorting it in ascending order.