1

Programming Assignment

Uopeople

CS1103

Professor Mopelola Akinyemi

05/05/2021

In regards to my work on fixing the bugs, I did not remove the code that caused the bug. Therefore, you will still see the bug code in my solution, however, I commented it out so it does not effect the solution.

Bug one, was found in the static method contains(). I placed a break point at the if statement to see what the code does. After analysing the code I realized that the else block of code was the cause of the problem. The reason the else block was the cause of the problem is simple. In Java, when we have a method with a return type, we must have a default return value in the method. This default return value is located at the end of the all the code in the method and it is also out side of any other block of code in the method. Since the default return value of this method is false, the if statements else block was checking if the value was in the array, if it was then it will return true else it will return false. This meant that every time the loop ran the increment of i never happened since the method just return false. It caused the method to never fully run through the the whole array.

Bug two, was found in the static method bubbleSort(). I tried to do this one with the debugger however, I was unable to find the problem. Nonetheless, I was able to find the bug just by looking at the construct of the inner for loop that was used for ‘j’. Instead of incrementing j, the loop incremented i. I was able to fix it just by making the inner for loop increment j instead of i.

Bug three, was found in the static method selectionSort(). I used the debugger again and realized that only two integer was ever sorted. That caused me to have a closer look at the algorithm, and I found problem was a type or a loss of concentration. The problem was in the if statement, instead of checking for i the algorithm only checked for array element at position one. I fixed the bug by removing the one and inserting i so that the algorithm and be dynamic with the for loop.

Bug four, was found in the static method insertionSort(). I used the debugger and realized that the while loop was checking up until n – 2 instead of checking until n – 1. This bug was fixed by letting the while loop run until pos >= 0 instead of pos > 0.

Bug five, was found in the static method insertionSort(). However, this bug was not in the while loop but rather at the end of the for loop. Again by running the debugger I was able to see that by assigning temp to pos caused the algorithm to run out of bound. Therefore, all I had to do to fix it was to assign temp to pos + 1 instead of pos.