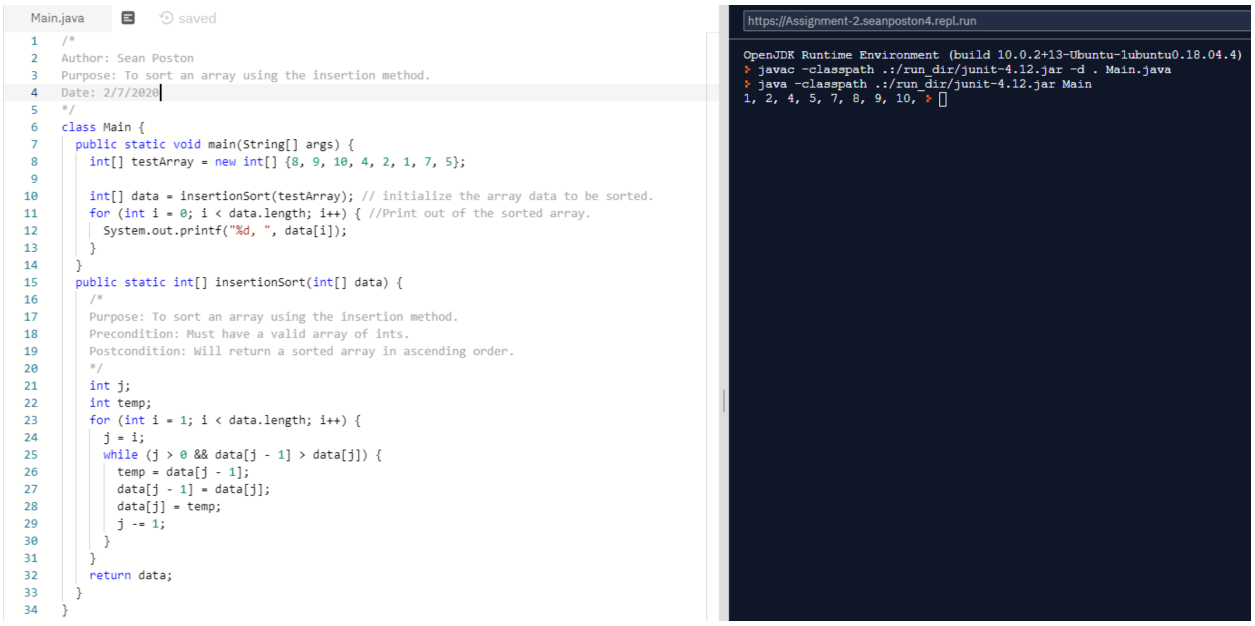
Sean Poston

Assignment 2

https://repl.it/@seanposton4/Assignment-2



```java
/*
Author: Sean Poston
Purpose: To sort an array using the insertion method.
Date: 2/7/2020
*/
class Main {
  public static void main(String[] args) {
    int[] testArray = new int[] {8, 9, 10, 4, 2, 1, 7, 5};

    int[] data = insertionSort(testArray); // initialize the array data to be sorted.
    for (int i = 0; i < data.length; i++) { //Print out of the sorted array.
      System.out.printf("%d, ", data[i]);
    }
  }
  public static int[] insertionSort(int[] data) {
    /*
    Purpose: To sort an array using the insertion method.
    Precondition: Must have a valid array of ints.
    Postcondition: Will return a sorted array in ascending order.
    */
    int j;
    int temp;
    for (int i = 1; i < data.length; i++) {
      j = i;
      while (j > 0 && data[j - 1] > data[j]) {
        temp = data[j - 1];
        data[j - 1] = data[j];
        data[j] = temp;
        j -= 1;
      }
    }
    return data;
  }
}
```

```
OpenJDK Runtime Environment (build 10.0.2+13-Ubuntu-1ubuntu0.18.04.4)
› javac -classpath .:/run_dir/junit-4.12.jar -d . Main.java
› java -classpath .:/run_dir/junit-4.12.jar Main
1, 2, 4, 5, 7, 8, 9, 10, ›
```

An insertion sort, in my opinion, is much straighter forward than a bubble sort. An insertion sort will go index by index and move the original element in that index toward the front of the array until it is properly sorted.

To walk through part of the example at hand:

1. The array is sent to the function insertionSort to be sorted.
2. The for loop takes the element at array[1] and compares it to the item before it (array[0]).
3. If array[1] < array[0], the program will swap the two using the temp variable, if not it will move to the next element.
4. The for loop then moves to the next element, array[2], and compares it to array[1].
5. If array[2] < array[1], the program will swap the two. Assuming yes, it will then check the new array[1], which was previously array[2], and compare it to array[0] and swap if necessary.
6. The program will repeat these steps for however many elements there are until the array is sorted.