Insertion Sort

The Insertion Sort Algorithm below sorts an array of integers into ascending order as follows:

1. Loop from `j = 1` to `j = elements.length−1` inclusive, completing `elements.length−1` passes.
2. In each pass, move the item at index `j` to its proper position in `elements[0]` to `elements[j]`:
    a. Copy item at index `j` to `temp`, creating a "vacant" element at index `j` (denoted by `possibleIndex`).
    b. Loop until the proper position to maintain ascending order is found for `temp`.
    c. In each inner loop iteration, move the "vacant" element one position lower in the array.
3. Copy `temp` into the identified correct position (at `possibleIndex`).

At the end of each pass, items at `elements[0]` through `elements[j]` are in ascending order.

```
/**
 * Sort an array of integers into ascending order.
 *
 * @param elements an array containing the items to be sorted.
 *
 * Postcondition: elements contains its original items and items in elements
 *                are sorted in ascending order.
 */
public static void insertionSort(int[] elements)
{
   for (int j = 1; j < elements.length; j++)
   {
      int temp = elements[j];
      int possibleIndex = j;
      while (possibleIndex > 0 && temp < elements[possibleIndex - 1])
      {
         elements[possibleIndex] = elements[possibleIndex - 1];
         possibleIndex--;
      }

      elements[possibleIndex] = temp;
   }
}
```