Homework Turnin

Email:	rgalanos@fcps.edu
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	03-02
Receipt ID:	d2a5c9845c916e6dc5b84882164c60ad

Turnin Successful!

The following file(s) were received:

Search_Driver.java (2750 bytes)

```
   //name:    date:
import java.io.*;      //the File
import java.util.*;    //the Scanner
import javax.swing.*;  //the JOptionPane
public class Search_Driver  {
   public static void main(String[] args) throws Exception
   {
      Scanner keyboard = new Scanner(System.in);
      System.out.println("Filename?");
      Comparable[] array = input(keyboard.next());
      //for(Comparable x: array)
      //   System.out.println(x);

      System.out.println("What word do you want to find?");
      String word = keyboard.next();
      int loc1 = Searches.linear(array, word);
      int loc2 = Searches.binary(array, word);
      System.out.println("Linear Search found it at location " + loc1 + " in " + Searches.linearCount + " comparisons.
      System.out.println("Binary Search found it at location " + loc2 + " in " + Searches.binaryCount() + " comparison

   }
   public static String[] input(String filename) throws Exception
   {
      Scanner infile = new Scanner(new File(filename));
      int count = 0;
      while(infile.hasNext())
      {
         infile.next();
         count++;
      }
      Scanner infile2 = new Scanner(new File(filename));
      String[] array = new String[count];
      for(int i=0;i<array.length;i++)
      {
         array[i] = infile2.next();
      }

      for(int x=1;x<array.length;x++)
      {
         String store = array[x];
         int y = x-1;
         while(y>=0 && store.compareTo(array[y])<0)
         {
            array[y+1] = array[y];
            y--;
         }
         array[y+1] = store;
      }
      return array;
   }
}
```

```
   ////////////////////////////////////////////////////
class Searches
{
   public static int linearCount = 0;
   private static int binaryCount = 0;

   public static int binaryCount()
   {
      return binaryCount;
   }
   public static int linear(Comparable[] array, Comparable target)
   {
      for(Comparable x: array)
      {
         linearCount++;
         if(x.compareTo(target)==0)
            return linearCount-1;
      }
      return -1;
   }
   public static int binary(Comparable[] array, Comparable target)
   {
      return binaryhelper(array, target, 0, array.length-1);
   }
   private static int binaryhelper(Comparable[] array, Comparable target, int start, int end)
   {
      int middle = (start +end)/2;
      binaryCount++;
      if(start>end)
         return -1;
      else if(array[middle].compareTo(target)==0)
         return middle;
      else if(array[middle].compareTo(target)<0)
         return binaryhelper(array, target, middle+1, end);
      else if(array[middle].compareTo(target)>0)
         return binaryhelper(array, target, start, middle-1);
      return -1;
   }
}
```