```
import java.io.*;
import java.util.*;
/**
 * Driver04 creates Weight Comparables from a text file then sorts them
 *
 * @author  Nathan Chen
 * @version 3-19-19
 * @teacher Coglianese
 * @period  2
 */
public class Driver04
{
    /**
     * Main executes upon run
     *
     * @param   args        Ignore
     * @throws  Exception   Catches FileNotFound
     */
    public static void main(String[] args) throws Exception
    {
        Comparable[] array = input("data.txt");
        sort(array);
        output(array, "output.txt");
    }

    /**
     * Creates an array of comparables from a file
     *
     * @param   filename    String for name of file to read
     * @return              Array of Comparables with data from file
     * @throws  Exception   Catches FileNotFound
     */
    public static Comparable[] input(String filename) throws Exception
    {
        Scanner infile = new Scanner( new File(filename) );
        int numitems = infile.nextInt();
        Comparable[] array = new Weight[numitems];
        for(int k = 0; k < numitems; k++)
        {
            array[k]=new Weight(infile.nextInt(),infile.nextInt());
        }
        infile.close();
        return array;
    }

    /**
     * Outputs any array of objects to a file
     *
     * @param   array       Array of Objects to output to file
     * @param   filename    String for name of output file
     * @throws  Exception   Catches FileNotFound
     */
    public static void output(Object[] array, String filename) throws Except
ion
    {
        PrintStream s = new PrintStream(System.out);
        System.setOut(new PrintStream(new FileOutputStream(filename)));
        for(int k = 0; k < array.length; k++){
            System.out.println(array[k].toString());
        }
        System.setOut(s);
    }

    /**
     * Sorts array of Comparables using selection sort
     *
     * @param   array   Comparable array to sort
     */
    public static void sort(Comparable[] array)
    {
        int maxPos;
        for(int k = 0; k < array.length; k++)
        {
            maxPos = findMax(array, array.length - k);
            swap(array, maxPos, array.length - k - 1);
        }
    }

    /**
     * Finds the largest Comparable in a section of an array using the compa
reTo method
     *
     * @param   array   Array of Comparables to search through
     * @param   index   Integer index of last element in section
     * @return          Gets the index of the largest Comparable
     */
    public static int findMax(Comparable[] array, int index){
        int temp = 0;
        for(int i=0;i<index;++i){
            if(array[temp].compareTo(array[i])==-1){
                temp=i;
            }
        }
        return temp;
    }

    /**
     * Helper method to swap two elements in a Comparable array
     *
     * @param   array   Array of Comparable to swap elements
     * @param   a       Integer index of first element to swap
     * @param   b       Integer index of second element to swap
     */
    public static void swap(Comparable[] array, int a, int b){
        Comparable temp = array[a];
        array[a]=array[b];
        array[b]=temp;
    }
}
```