C:\Users\Rich\Documents\NetBeansProjects\Lab12\src\NameComparator.java

```java
1
2  /**
3   * This comparator is based on the Lexicographic representation of a string.
4   * @author Richelin Metellus
5   * @version 04/21/2017
6   */
7  public class NameComparator implements Comparator<Voter> {
8
9      @Override
10     public int compare(Voter a, Voter b) throws NullPointerException
11     {
12         if( a == null || b == null)
13             throw new NullPointerException("Invalid object reference");
14
15         int comparisonResult = 0;
16
17         String aName = a.getName();
18         int aNameSize = aName.length();
19         String bName = b.getName();
20         int bNameSize = bName.length();
21
22
23  // finding lesser string length to lower # of loop iterationn
24         int lengthDiff = aNameSize - bNameSize;
25         int iterationLimit = (lengthDiff  >= 0) ? bNameSize: aNameSize;
26
27         for ( int i = 0; i< iterationLimit; ++i)
28         {
29             char aNameCur = aName.charAt(i);          // current char testing for nameA
30             char bNameCur = bName.charAt(i);          // current char testing for name of Voter b.
31             if(aNameCur < bNameCur)              // a = 97 < b =98 in ASCII value
32             {
33                 comparisonResult = 1;
34                 return comparisonResult;
35             }            // i.e aName comes 1st alphabeically
36             else if( aNameCur > bNameCur)
37             {
38                 comparisonResult = -1;
39                 return comparisonResult;
40             }
41             else if (aNameCur == bNameCur &&aNameSize==bNameSize){
42                 comparisonResult = 0;
43
44             }
45
46             else if (aNameCur == bNameCur &&aNameSize < bNameSize )  // special case.
47                 comparisonResult = 1;                              // if comparing Ama with Aman
48             else if (aNameCur == bNameCur &&aNameSize > bNameSize)  // special case.
49                 comparisonResult = -1;
50         }
51         return comparisonResult;           // if loop fully executes and reaches this point it return 1 or -1,0.. for sepcial case
52
53     }
54
55     //----------------Private utility --------------------------
56
57     protected static char[] stringtoCharArray(String S)
58     {
59         char[] charArray = new char[S.length()];
60         for(int i = 0; i < S.length(); ++i)
61         {
62             charArray[i] = S.charAt(i);
```

```
63        }
64        return charArray;
65    }
66 }
67
```