**Presidents Lab**

Interfaces and Sorting using Comparable and Comparator

**Summary**

You will write a President class which implements the Comparable interface. You will write a driver class to create an ArrayList of President object by reading in a comma separated file. You will use the driver class to test your methods. You will write two classes that implement the parameterized Camparator<T> interface.

**President Class**

Create a President class with the following traits:

1. Add the implements statement required for the Comparable interface (non-parameterized)
2. State Variables: String lastName, String firstName, String middle Name, int orderNumber, String homeState
3. Constructor with parameters used to assign each instance variable
4. Public accessor methods to get each instance variable
5. Public getFullName method returns a string in the form: Last, First Middle
6. Override the equals and toString method inherited from the Object class.
   1. equals will evaluate the presidents’ full names
   2. toString will return a formatted string with all instance variables:
      1. Full name order number home state
      2. left-justified, with the following widths 20, 3, and 12 characters
7. Implement the compareTo(Object obj) method. The return value is based on the order number.

**Tester Class**

Create a Tester class to test the President class and others.

1. Add a main method. Main will create an ArrayList and fill it with President objects.
2. Add a readFile method that takes in a string for the file name and an ArrayList to fill. This method will read in the presidents.csv file and use it to create President objects and fill the ArrayList.
3. Use main to test the President class’s compareTo methods, by sorting and looking at the results.

**PresidentNameComparator and PredidentHomeState Classes**

Create the two classes that implement the parameterized Comparator interface. For each class:

1. Add the implements statement required for the Comparator interface
2. Implement the equals and compare methods. The return value of the compare method will be based on the name (alphabetical order) and the home state (alphabetical) respectively.