

Alien Database

1. Checkout the project called “131Fall18Lab25”
2. Complete the Alien class. Note that this class implements the Comparable interface. At the top it says “Comparable<Alien>” – we haven’t talked about that syntax, but this is the correct way to implement the Comparable interface using generics.
 - a. Familiarize yourself with the code that has been provided.
 - b. Implement the static method called “reverseString”. It should return a String that is the reverse of the parameter. For example, if the parameter is “monkey”, then the return value should be “yeknom”.
 - c. Test your reverseString method using the JUnit test called “testReverseString” that has been provided.
 - d. Implement the compareTo method.

To compare the aliens you must compare the reverse spellings of their names.
For example: An alien named “ZARA” would be considered “less than” an alien named “FAWZI” because “ARAZ” would come before “IZWAF” in the dictionary.
Follow the usual contract for compareTo. (Ask your TA if you’re not sure about it.)
 - e. Test your compareTo by running the JUnit test called “testAlien” that has been provided.
3. Complete the AlienDatabase class:

The database will be an ArrayList of size 10. Each of the 10 entries in this list will be an ArrayList of aliens. (Think of it as 10 rows, where each row contains a list of aliens). In other words, you are creating an ArrayList of ArrayLists of Aliens.

Each alien has between 1 and 10 heads. The first row in the database is where we will store the aliens with 1 head. The second row is where we will store the aliens with two heads, etc.

Each row will be ordered according to the “natural order” of the aliens in that row. (That means according to the reverse alphabetical order of their names.)

 - a. Declare an instance variable called “data”. This variable’s type should be an ArrayList containing ArrayLists of Aliens. (I will let you figure out the syntax!)
 - b. Write a constructor with no parameters. It will instantiate the database and add 10 empty rows.
 - c. Write a method called addAlien, that takes an Alien as a parameter. It should:
 - Add the alien to the correct row based on the number of heads.
 - Sort the row using Collections.sort. (This will automatically use your compareTo method to sort them.)
 - d. Test your work by running the main method that has been provided.