

CS 351 Lab 2 (Due February 23rd 11:59 pm)

Prompt

Your task is to create a contiguous list implementation. This class behaves like a list to users but behind the scenes uses an array to store element. The array should initially have a length of 10. If the user adds more elements than the array can handle then you must create a new array with double the length of the previous length. For simplicity our list will only be able to hold Strings. You must implement the following methods:

- void prepend(String e)
 - Adds the given element to front of the list
- void append(String e)
 - Adds the given element to the end of the list
- void insert(int index, String e)
 - Inserts the given element add the given index
 - Shifts all elements after the given index up by 1 index
- void empty()
 - Removes all elements from the list
- boolean has(String e)
 - Returns true if the given element exists in the list, false otherwise
- String retrieve(int index)
 - Retrieves the element at the given index, if the index doesn't exist then return null
- boolean isEmpty()
 - Returns true if the list is empty, false otherwise
- String delete(int index)
 - Deletes the element at the given index and returns it, if the index doesn't exist then return null
- boolean delete(String e)
 - Deletes the first occurrence of an element from the list if it exists, if an element is removed return true, false otherwise
- boolean deleteAll(Collection<?> c)
 - Deletes all elements within the given collection, this includes duplicates. If it removes an element then return true, otherwise false
- String mutate(int index, String e)
 - Sets the element at the given index to the given element and returns the old element, return null if the index doesn't exist
- int length()
 - Returns the current length of the list
- String toString()
 - Override toString from Object
 - Should print out with a square bracket at the front and the back of the list
 - Each element should be seperated by a comma and a space

- For example if your list had the elements “1” “2” and “3” then the output would be “[1, 2, 3]”
- boolean equals(Object o)
 - Override equals from Object
 - Should return true if the contents of your array are equal to the contents of o’s array, otherwise false

You must use the given starter file and are not allowed to change any of the type signatures of the above functions. You are welcome (and encouraged) to add more methods and fields as you see fit. I have also given you a file with tests to see if your code is working.

Submission

Submit a zip of only your java source files onto canvas. No other files or folders should be within the zip file **ONLY** your java source files. You should not include the test file in your submission.