CS 314 Exam One Review — Insert All To Front — **Solution**

Array-based Lists

Implement an instance method for the `GenericList` class which will insert all of the elements from another `GenericList` to the front of `this` list. The `GenericList` argument will remain unchanged after this operation.

Complete the following method.

```
// Removes inserts all elements from other into this
// pre: other != null
// post: other is unchanged
public void insertAllFront(GenericList<E> other)
```

Here are some example calls to `insertAllFront()`:

```
4, 5, 6].insertAllFront([1, 2, 3]) → this = [1, 2, 3, 4, 5, 6]
[4].insertAllFront([3, 1]) → this = [3, 1, 4]
[2, 3, 4, 5].insertAllFront([1]) → this = [1, 2, 3, 4, 5]
[].insertAllFront([1, 2, 3]) → this = [1, 2, 3]
[3, 1, 1].insertAllFront([]) → this = [3, 1, 1]
```

Your method will be in the following `GenericList` class:

```
public class GenericList<E>{
  private int size;
  private E[] con;
  // ...
}
```

Do not use or assume there are any provided methods in the GenericList class.
You may create a new internal array container.
Do not use any other Java classes or methods.

```
// Removes inserts all elements from other into this
// pre: other != null
// post: other is unchanged
public void insertAllFront(GenericList<E> other){
  E[] newCon = (E[]) new Object[(this.size + other.size) * 2];
  int indexToAdd = 0;

  for(int i = 0; i < other.size; i++){
      newCon[indexToAdd] = other.con[i];
      indexToAdd++;
  }
  for(int i = 0; i < this.size; i++){
      newCon[indexToAdd] = con[i];
      indexToAdd++;
  }

  con = newCon;
  size = indexToAdd;
}
```

This is pretty straightforward problem where we just merge two arrays together into one larger array. The trickiest part is definitely the first line in the solution where we create a new array of Generic types. It is important to remember how to do this strange casting trick to create an array of E's. Also, this new array has to have a new, larger size. Your solution does not need to have the exact same resize as this solution, but it should leave extra capacity in the internal container. Otherwise, the very next add/insert operation for this list it guaranteed to be O(N). Also, in the two for loops, make sure you iterate up to the size of the lists, not the length of their internal array contianers. Finally, make sure to update the two instance variables in the list before your method returns. These two can be very easy to forget but are extremely important.