

Junior Knights: ArrayList of Predefined Object exercises

Useful ArrayList Methods

```
// Adds e to the end of this ArrayList.
boolean add(E e)

// Adds e to position index of this ArrayList (sliding rest of
// the items over to the right one slot.)
void add(int index, E e)

// Empties this ArrayList.
void clear()

// Returns true iff this ArrayList contains o.
boolean contains(Object o)

// Returns the element at position index of this ArrayList.
E get(int index)

// Returns the minimum index in this ArrayList that contains o.
// If there is no such index, -1 is returned.
int indexOf(Object o)

// Removes and returns the item at position index of this
// ArrayList.
E remove(int index)

// Removes the first occurrence of o from this list, if present
// and returns true in this case. Returns false otherwise.
boolean remove(Object o)

// Sets the item at position index to e and returns the old item
// stored in this position.
E set(int index, E e)

// Returns the size of this ArrayList.
int size()
```

Useful Collections Methods

```
static E max(Collection c)
static E min(Collection c)
static void reverse(List list)
static void shuffle(List list)
static void sort(List list)
```

- 1) Write a program that reads in several grocery items(just of type string), terminated by the string "end", and then asks the user to for items to search for and says whether or not each item is in stock or not.
- 2) Write a program that reads in several grocery items (terminated by the string "end"), then reverses the list manually (actually change which index each item is stored), and then prints out this resultant list. Do this WITHOUT the function Collections.reverse(List<?> list).
- 3) Write a program that reads in a sorted list of grocery items that may contain repeats and removes the repeats.
- 4) Write a program that allows the user to add items, remove items and print the items in their grocery cart. The program should continue until the user wants to quit. If the item to be removed isn't in the cart, an error message should be printed explaining that the item could not be removed since it wasn't in the cart to begin with.
- 5) Populate an ArrayList of Integer with the values 1 through 100. Shuffle this list a million times and count how many times the first item in the list is equal to 1. Think of other things to keep track of over the million shuffles and print out that information after the simulation.
- 6) Simulate the lottery - fill an ArrayList of Integer with numbers 1 to 52. Shuffle and remove the item at index 0 6 times. Take these six items, sort them and print out the winning combination.
- 7) Go to: <https://docs.oracle.com/javase/7/docs/api/java/util/Collections.html>. Play around with some of the different methods in the Collections class. Here are some that I think are cool:

```
disjoint  
frequency  
max  
min  
replaceAll  
rotate  
shuffle  
sort  
swap
```

Note: The AP exam doesn't allow the use of most of the Collections methods. Keep that in mind. But, when you're designing interesting programs, having methods such as these at your disposal reduces bugs and the time you spend completing your programs, and also tends to allow you to design better programs. (This is why we use APIs to begin with!!!) So, I think it's great to start learning how to get familiar with pre-written classes as it's one of the most practical skills for any programmer.