CSCI-101 Programming 1

Lab 17

INSTRUCTIONS

I. Create a directory named **lab17** in your **labs** directory.

II. Copy your **PickupTruck.java** file and your **TruckInventoryApp.java** file from your **lab16** directory into **lab17** directory.

III. Update your **TruckInventoryApp** class so that it satisfies the following (changes from lab15 are in **blue**):

A. Create a method named **printMenu** that prints to the screen the following menu:

1) Import trucks
2) Print trucks
3) Add truck
4) Remove truck
5) Export trucks
6) Quit

Choose option:

B. Create a method named importTrucks. The method asks the user for the name of a file and creates a scanner that can scan the file.

The method reads the first line to determine the number of entries in the file and allocates an array to hold the entries.

For each entry in the file, the method creates a PickupTruck and stores the truck in the array.

When finished reading the entries, the method returns the array.

C. Create a method named **printTrucks** that takes an array of **PickupTruck** elements as an argument and prints the Strings returned by **toString** for each truck in the array.

D. Create a method named **addTruck** that has an array of **PickupTruck** elements named **array** as the first parameter and an integer named **count** as a second parameter. The parameter **count** specifies how many elements are currently stored in the array.

If there is no room in the array for a new truck then expand the array.

The method asks the user for a VIN, make, and model, then reads the data from the keyboard, creates an instance of **PickupTruck** with the data that was read, and adds the truck to the array.

The method returns either the array that was passed in or the expanded array (if created).

E. Create a method named removeTruck that has an array of PickupTruck elements named array as a parameter. The method asks the user for a VIN, reads the VIN from the keyboard and removes the first instance of PickupTruck in the array that has that VIN and if found shifts all of the elements to the left so that there are no gaps in the array of elements. The method should return true if a truck was removed; otherwise it returns false.

F. Create a method named **exportTrucks** that has an array of **PickupTruck** elements named **array** as the first parameter and an integer named **count** as a second parameter. The parameter **count** specifies how many elements are currently stored in the array.

The method asks the user for a file name and creates a **PrintWriter** that can write to the file. The method then writes on the first line of the file the number of elements that are in the array (**count**). Then, for each element in the array, the method writes the VIN, make, and model to the file with commas separating them. Each truck should be written on a separate line of the file.

G. In **main** declare an array named **inventory** that can hold 20 **PickupTruck** elements and an integer named **count** that holds the number of elements currently stored in the array. Then repeatedly display the menu, read the value entered by the user, and perform the appropriate action.