**Java Software Development**

**Laboratory I**

*The purpose of this laboratory is to review the use of interfaces as part of OO programming.*

Using the Comparable interface, Cloneable interfaces, copy constructor and the shallow-deep clone (copy) concept to create a driver record application, described as follows.

Note: You should practise both copy constructor and Cloneable.

**Download and study codes for advanced shallow and deep copy example and study them in depth.**

Write these classes with necessary methods similar to example codes.

Driver class

* Name (first and last name)
* Driver ID
* Address
* List of Cars

Address class

* Street number
* Street name
* PostCode
* Country

PostCode class

* Suburb
* City
* State

Car class

- int id;

- String model;

- double price;

- CarType carType // an enum { SUV, Sedan, Sport, Luxury, etc}

Create an ArrayList of drivers in your main and add three drivers to the list (test code). Add some cars to each Driver by using addCar() method.

Make three empty ArrayLists one for shallow copy, one for deep copy with copy constructor and one for deep copy with clone().

Write 3 methods for shallow and deep copies of a list.

Add compareTo() to the Driver based on the city name in descending order

Add compareTo1() to the Driver based on the total value of the cars he owns. This can be replaced with the other compareTo() if required.

Develop a console application that has 5 options.

Option 1 is to get all the information from the user and creates a Driver object and adds it to the original list. Note: You need to keep asking for cars and add them to the driver.

Option 2 is to make a shallow copy, deep copy by using copy constructor, and deep copy by using clone() of original list, store them and then sorts the original list by city name in descending order.

Option 3 is to print original, shallow copy and deep copy lists on console.

Option 4 is to ask the user to supply a Driver ID and a city name and then finds the given driver in the original list and copies the driver record to a new record (using clone()). Then changes the city name of the driver and prints both objects. If driver ID is not found a proper prompt should be displayed.

Option 5 is to ask the user to supply a Driver ID as well as the information for a new car and then finds the given driver in the original list and copies the driver record to a new record (using copy constructor). Then adds the car to the driver record and prints both objects. If driver ID is not found a proper prompt should be displayed.