# Algorithms & Data Structures :

# Lab 9: 25th February 2019

**Phone Directory Application.**

Classes:

PDApplication – has main method

PDGUI – the GUI class

PhoneDirectory – the interface for representing a directory

ArrayBasedPD – array implementation of PhoneDirectory interface

PhoneDirectoryEntry – a class representing one entry in the directory – why is String used for number

This application illustrates:

* separation of business logic from GUI logic
* use of an interface – here an array is used to implement the PhoneDirectory interface
  + What other implementations are possible?
* use of Arrays class sort and binarySearch methods

A list of names and phone numbers are read in from a file. The file has a name on one line and the number on the following line. These details are stored in an array. This is a partially filled array. It is sorted based on name.

The application allows you to view the entries, to add a new entry, search on name, save to a file.

Sort and search functions are required here. You have written code for sorting algorithms and binary search. Here, you should use the sort and binary search in Arrays class.

The Add, Search and Save Directory buttons have no functionality behind them.

For this lab:

1. Update the PhoneDirectoryEntry class to implement Comparable so that the entries can be sorted by name. Then update the code so that the array is sorted after the entries have been read in from the file. Note the Arrays.sort method you call is the one for a partial array, not the one for a complete array i.e.

public static void sort​([Object](https://docs.oracle.com/javase/10/docs/api/java/lang/Object.html)[] a,

int fromIndex,

int toIndex)

Sorts the specified range of the specified array of objects into ascending order, according to the [natural ordering](https://docs.oracle.com/javase/10/docs/api/java/lang/Comparable.html) of its elements. The range to be sorted extends from indexfromIndex, inclusive, to index toIndex, exclusive.

1. Implement the Previous button. It will be similar to code for Next button.
2. Implement the Add option. Ask the user to enter name and number and add a new entry to the end of the array. The array should be sorted again.
3. Implement the Search buttons. Ask the user to enter a name. The application should then search the array and display a message box with corresponding number or a suitable message if the name is not found.
4. The save button can also be implemented. Write the data back to the file. You could check first if the data has been updated, and only write back to the file if it has.