M – Arrays – Sorted

Student Exercises

# Exercises

1. **PhoneBook** – [Sorting by Name and Sorting by Phone Number] This extends the PhoneBook class by adding sorting by phone number and a binary search.
2. **Registrar** – [Sorting by ID] The Registrar class is responsible to support the enrollment of students. This class maintains the student body as a collection of Student objects. It supports the ability to find and remove students, switch students to another program, and get the number of students enrolled in a specific program.
3. **CDLibrary** – [Not Yet Implemented] This class manages a collection of CDs that are stored in alphabetical order. It supports searching the library for CDs based on the title.

# PhoneBook

## Problem Statement

Extend the PhoneBook class by adding sorting by name and a binary search when sorted by number.

* **FindPhoneNumberUsingBinarySearch()** – This private method will do a binary search of the PhoneNumber array to find a PhoneNumber with a matching telephone number.
* **BubbleSortByLastNameThenFirstName()** – This private method sorts the array by first name within last name, in ascending alphabetical order.

## Supporting Classes

# Registrar

## Problem Statement

[Sorting by ID] The Registrar class is responsible to support the enrolment of students. This class maintains the student body as an array of Student objects. It supports the ability to find and remove students, switch students to another program, and get the number of students enrolled in a specific program. Code the following methods to complete the Registrar class.

* **RemoveStudent()** – This method must be modified to maintain the sort order when removing a student from the array.