

```
import java.util.Scanner;

/**
 * COSC311 – Program 1
 *
 * This is the driver for Program 1, a database implementation.
 * This program reads in data from an external file of 68 records.
 * Each record is composed of three fields:
 *     (String lastName) (String firstName) (String ID)
 *
 * This menu driven command line program offers options to add a
 * record to the database, delete a record, search for a record
 * and print out the entire database in different orders.
 *
 * @author Bill Sverdlik, Mordechai Sadowsky
 * @version Version 1.0, 02-feb-2014
 */

public class COSC311Driver {

    private static Scanner keyboard;
    private static DataStructure myStructure;

    public static void main(String[] args) {
        keyboard = new Scanner(System.in);
        myStructure = new DataStructure();
        int response;

        System.out.println("Welcome to YourStudentRoster");
        do {
            System.out.println("\nMain menu:\n");
            System.out.println(" 1 Add a new student");
            System.out.println(" 2 Delete a student");
            System.out.println(" 3 Find a student by ID");
            System.out.println(" 4 List students by ID increasing");
            System.out.println(" 5 List students by first name increasing");
            System.out.println(" 6 List students by last name increasing");
            System.out.println(" 7 List students by ID decreasing");
            System.out.println(" 8 List students by first name decreasing");
            System.out.println(" 9 List students by last name decreasing\n");
            System.out.println(" 0 End");
            System.out.print("\nMenu selection: ");

            response = keyboard.nextInt();
            keyboard.nextLine();

            switch (response) {
                case 1: addIt();
                    break;
                case 2: deleteIt();
                    break;
                case 3: findIt();
                    break;
                case 4: myStructure.listIt(1, 1);
                    break;
                case 5: myStructure.listIt(2, 1);
            }
        } while (response != 0);
    }
}
```

```

        break;
    case 6: myStructure.listIt(3, 1);
        break;
    case 7: myStructure.listIt(1, 2);
        break;
    case 8: myStructure.listIt(2, 2);
        break;
    case 9: myStructure.listIt(3, 2);
        break;
    default:
    }
} while (response != 0);

System.out.println("\nThank you, goodbye!");
}

/**
 * Menu option 1: add a new student to the database.
 */
public static void addIt() {
    String name1, name2, tempID;
    boolean found;

    do {
        System.out.print("Enter a unique ID number to add: ");
        tempID = keyboard.nextLine();

        //is it unique?
        found = (myStructure.search(tempID) > -1);
        if (found) {
            System.out.println("ID already in use.");
            System.out.print("Please re-enter a unique ID: ");
        }
    } while (found);

    // We found a unique ID. Now ask for first and last name
    System.out.print("Enter first name: ");
    name1 = keyboard.nextLine();
    System.out.print("Enter last name: ");
    name2 = keyboard.nextLine();
    System.out.println();

    // add to our data structure
    if (!(myStructure.insert(name1,name2,tempID)))
        System.out.println("Error, database full!");
}

/**
 * Menu option 2: delete a student from the database.
 */
public static void deleteIt() {
    String tempID;
    boolean found;

    do {
        System.out.println("\nEnter the ID number of student to delete: ");

```

```
        tempID = keyboard.nextLine();

        //is it in the database?
        found = (myStructure.search(tempID) > -1);
        if (!found) {
            System.out.println("ID not found.");
            System.out.print("Please re-enter an ID to delete: ");
        }
    } while (!found);

    myStructure.delete(tempID);
}

/**
 * Menu option 3: find a student
 * If found, prints out the record.
 */
public static void findIt() {
    String tempID;
    boolean found;
    int recordNumber;

    do {
        System.out.println("\nEnter an ID number: ");
        tempID = keyboard.nextLine();

        //is it in the database?
        recordNumber = myStructure.search(tempID);
        found = (recordNumber > -1);
        if (!found) {
            System.out.println("ID not found.");
            System.out.print("Please re-enter an ID to find: ");
        }
    } while (!found);

    myStructure.print(recordNumber);
}
}
```