

COMPUTER PROGRAMMING

LAB ASSIGNMENT – 10

NOTES:

- i) Carefully read all assignments and there is no choice.
- ii) Each problem in this assignment has to be answered in the **same c file**.
- iii) Strictly follow proper name/style convention. When you are ready, submit the solution via google classroom.

iSiMS

For this assignment we will be creating a simple software called IIITS Student-Information Management System (iSiMS) . Please start by creating a structure (Struct) to capture the following student information.

- (1) First name
- (2) Last name
- (3) Roll no (should be unique)
- (4) CGPA
- (5) Courses registered (integer array – assume course-id is int)

The maximum number of students that the system should be able to handle is 50

[Hint: This means, it's OK to declare the students array as a global variable].

Please write **separate functions** to support each of the following operations. You must **unify** the below solutions into a single coherent software (**1 program**). The software runs **forever**, unless the user wants to quit.

[Hint: we use infinite-loop and switch-case in main for the desired effect]

You can write your own loop. To make it easier for you, I have attached a simple **template.c** file. You can use that as the base and build the solution on top of it. The options are presented as a numeric menu item, from which the user can choose from. The user can choose '0' to exit.

2 Marks for creation of the structure. Each function below gets 2 marks each. Strictly No partial marks.

- 1) addStudent: Get data from the user and add a new student to the list of students. (check for uniqueness of the rollno while adding. If the rollno is not unique, inform the user to try again)

- 2) findStudent (by rollno): Find the student record for the given roll no and print the details. (if not found, inform the user)
 - 3) findStudents(by first-name): Find all students with the given first name and print their details. (if not found, inform the user)
 - 4) findStudents(by course-id): Find all students who have registered for a given course (if not found, inform the user)
 - 5) totalStudents: print the total number of students in the system
 - 6) deleteStudent(by rollno): delete the student record for the given roll no [Hint: this requires some thinking and modify addStudent appropriately]
 - 7) updateStudent(by rollno): provides a way to update student record (should not ask for new details for all fields, the user should be able to pick and choose what they want to update. You can do this by providing a numeric menu for allowing the user to select the item that they want to modify.)
 - 8) PrintStudentStatistics: compute mean, median, std-dev of students CGPA and print it.
 - 9) PrintCourseSummaries: print a summary of how many students have registered in each course. Example:
 Course no 101 Total Students 31
 Course no 123 Total Students 17
 ...
 ...
-