

**Instructor:**

Mr. Samyan Qayyum Wahla

**CLO:**

- CLO3

**Registration Number:****Name:****Guidelines/Instructions:**

- Use of VS Code is must in this lab.
- Write well commented code.
- Name of variables should be meaningful.
- Code should be well formatted.
- Create meaningful variable names. Add comments for readability. Indent each line of your code.
- Plagiarism/Cheating is highly discouraged by penalizing to both who tried and one who shared his/her code.
- Make sure all validity operations are completed before calling add/update/delete/view/register functions.
- Comply with the given number of inputs in functions.
- Do not take input from user in functions

## University Learning Management System

### Overview of Project

You are required to develop a University Learning Management system. Every student enrolled in the university will have account on your Learning Management System. Account will contain Courses enrolled for a student, Grade Book (showing the GPA and CGPA of student), Account Book (providing the fee system), and Profile of the Student. Profile of the student will contain personal information of the student and status of the student in the university. Every student will have a unique ID and password to login on your Learning Management system. Administrator can add a new course. Administrator can print all Data of the student.

### Phase 3

Objectives of phase 3 of project are as following:

- Student Management Module
- Courses Registration for the student

### How to Start with Phase 3

The code of the phase 2 is provided at <http://shorturl.at/chlQY>. You have to use the provided code as baseline for the implementation of the phase 3. There is room available for improvements in the code of phase 2. You can choose to improve it, but you must provide the details if you make any modifications in the existing features.

### Tasks

1. Students will be managed with the following attributes

- a) Name of Student (can contain alphabets and space only with max length of 40)
- b) Registration Number of Student (make sure to keep it valid according to UET valid registration number)
- c) Courses of the student

2. Create the file of the students.txt which contain the data in the following format

```
2020-CS-564,Muhammad Ali, CS102:CS103
2020-CS-543,MuhammadAli, CS102:CS101:CS402
```

3. Create the following arrays in main function

- a) stdNamesList : Text
- b) stdRegNoList: Text
- c) stdCourseList: Text

4. create the variable in the main function with the name of MAX\_STUDENTS

5. Create the following functions and note \* section.

- a. **bool** addStudent(stdNamesList:Text Array, stdRegNoList: Text Array, stdArraySize: integer, studentName: Text, regNo:Text)
- b. **bool** updateStudent(stdNamesList:Text Array, stdRegNoList: Text Array, stdArraySize: integer, stdRegNoToUpdate:Text, studentName: Text, regNo:Text)
- c. **bool** deleteStudent(stdNamesList:Text Array, stdRegNoList: Text Array, stdCourseList: Text Array, stdArraySize: integer, regNo:Text)
- d. **bool** registerCourse(stdRegNoList: Text Array, stdCourseList: Text Array, courseCodeList: Text, , stdArraySize: integer, regNo:Text, courseCode:Text)  
This function checks if given courseCode is in courseCodeList or not. If yes, then adds it in stdCourseList against student's index.
- e. **bool** unRegisterCourse(stdRegNoList: Text Array, courseList: Text Array, stdArraySize: integer, regNo,Text, courseCode:Text)
- f. **bool** isValidRegistrationNumber(regNo:Text) – Acceptable Format 20XX-CY-001
- g. **bool** isValidStudentName(regNo:Text) – Use same rules as per course name.
- h. **bool** loadStudents()
- i. **bool** saveStudents()

**Note:** As the courses of the students can be multiple, if the student is registered in CS102 and CS103 then the stdCourseList index of that student will have the value CS102:CS103

6. Update the menu to handle student management

Choose the following option

- 1 Add Course
- 2 Update Course
- 3 Delete Course
- 4 View All Courses
- 5 View Courses of a Semester
- 6 Add New Student
- 7 Update Student
- 8 Delete Student
- 9 View All Students
- 10 Register the course for student
- 11 Unregister the course for student

12 Logout of the system

13 Exit Program

7. If the user opts to add student, show the following details and take input from user

Enter details of Student: 2020-CS-564 Muhammad Ali

Student has been added successfully

8. If the user opts to edit student, show the following details and take input from user

Enter registration Number of the student to edit: 202--CS-564

Enter details of Student: 202--CS-564 Muhammad Ali Ahmed

Student has been edited successfully

9. If the user opts to delete student, show the following details and take input from user

Enter registration Number of the student to delete: 2020-CS-564 Student has been deleted successfully

10. If the user opts to register course for student, show the following details and take input from user

Enter registration Number of the student for course registration: 2020-CS-564

Enter Course Code to register: CS101

Course has been registered for student

11. If the user opts to unregister course for student, show the following details and take input from user

Enter registration Number of the student: 2020-CS-564

Enter Course Code to unregister: CS101

Course has been unregistered