

W e b E n g i n e e r i n g L a b

Lab 01**Marks 100**

Instructions

Work on this lab individually.

You are **NOT** allowed to use the internet, or mobile phone.

You are **NOT** allowed to borrow anything from your peer student.

What you have to do

Program the following tasks. The name of your files will be according to the task given in this lab.

Task 1**[100]**

You are supposed to build a Record System for our class with following activities:

- View record
- Add a new record
- Update existing record
- Delete existing record

Create a class **Student** with attributes ID/roll-no, name, age, gender, and CGPA.

Your program must ask for the total number of students in a class as an input, and create an array to contain their records. Display a menu (using switch and infinite-loop) to add, view, delete or exit.

int addStudent(): Input the required data and store it into next empty index of array. The ID/roll-no must be the index of array.

Your program must return the ID/roll-no (index) where data is stored or -1 (as an error message) if no space is available.

viewStudent(): Your program must ask for the ID/roll-no and show the particular record (index). Display an error message accordingly.

deleteStudent(): Your program must ask for the ID/roll-no and free the particular record (index). Display an error message accordingly.

updateStudent(): Your program must ask for the ID/roll-no and update the particular record (index). Display an error message accordingly.

Write a **main()** to show the execution of your code.

Instruction:

- Create an array of objects to store the Students.
- Use GUI components for input and output.
- Convert the string arguments into appropriate types using the concept of wrapper classes. e.g., from string to int conversion:
`int num = Integer.parseInt("123");`
- For exit, you may call to `System.exit(0)`
- The syntax of conditioning and looping is the same as you used in C++.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺
