



# HARAMAYA UNIVERSITY

*We Build the Basis for Development*

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

COURSE TITLE: DATA STRUCTURE AND ALGORITHMS

COURSE CODE: SENG2032

---

***STUDENT REGISTRATION SYSTEM USING SINGLY  
LINKED LIST***

---

<u>GROUP MEMBERS</u>	<u>ID NUMBER</u>
----------------------	------------------

- |                           |         |
|---------------------------|---------|
| 1. ELIAS ASEFA.....       | 0917/13 |
| 2. YADUMA LECHISA.....    | 2817/13 |
| 3. FIRISAT DESSALEGN..... | 1157/13 |
| 4. GEZAGN LEMI.....       | 1260/13 |
| 5. KALEB ALEBACHEW .....  | 1593/13 |

ELIAS ASEFA  
[COMPANY NAME]  
[Company address]

## SYSTEM REQUIREMENTS (FUNCTIONALITIES)

1. Ability to register or add student to record.
2. Ability to display registered students.
3. Ability to modify or update student information.
4. Functionality to delete existing student records.
5. Functionality to search registered student from the record.
6. Functionality to sort student information based on name, GPA, or ID.
7. Functionality to exit program incase when needed.
8. Ability to traverse across operations without closing the program.

## ALGORITHMS IMPLEMENTED TO BUILD THIS PROGRAM

1. Registering new student.
  - ✓ Prompt the user to enter all the student information one by one.
  - ✓ If new student to be registered is the only student in the record, put the student record at the starting (head) of list.
  - ✓ Else, place student on the last line since you are going to sort it based on different criteria.
  - ✓ If recording is successfully done, display reply as “registered successfully”.
  - ✓ Go back to main menu.
2. Display registered students to screen(console).
  - ✓ Check whether record is empty or not.
  - ✓ If record is empty, display warning “Empty list”.
  - ✓ Otherwise display all the record data.

### 3. Delete existing student information.

- ✓ Check whether record is empty or not.
- ✓ If record is empty, display warning "Empty list".
- ✓ If record is not empty, prompt the user to enter student ID to identify student record to be deleted.
- ✓ If student to be deleted is found in the record delete its information and respond to confirm operation is successful.

### 4. Search student by his/her ID number.

- ✓ Check whether record is empty or not.
- ✓ If record is empty, display warning "Empty list".
- ✓ If record is not empty, prompt user to enter ID number of student then check all the record data if match with entered ID is found.
- ✓ If found, declare student is found in the record
- ✓ Otherwise warn student does not found.

### 5. Sort student record by name.

- ✓ Check whether record is empty or not.
- ✓ If record is empty, display warning "Empty list".
- ✓ If record is not empty, prompt user to enter ID number of student, then check all the record data if match with entered ID is found.
- ✓ If match is found, do the following operations.
  - If the nth student name is alphabetically less than the n-1th student, swap nth student with n-1th student.
  - Continue this operation until the whole record is sorted.

- If operation is finished without any problem, declare sorting is finished successfully.

#### 6. Sort student record by ID number.

- ✓ Check whether record is empty or not.
- ✓ If record is empty, display warning “Empty list”.
- ✓ If record is not empty, prompt user to enter ID number of student then check all the record data if match with entered ID is found.
- ✓ If match is found, do the following operations.
  - If the nth student ID is less than the n-1th student, swap nth student with n-1th student.
  - Continue this operation until the whole record is sorted.
  - If operation is finished without any problem, declare sorting is finished successfully.

#### 7. Sort student record by GPA (ascending order).

- ✓ Check whether record is empty or not.
- ✓ If record is empty, display warning “Empty list”.
- ✓ If record is not empty, prompt user to enter ID number of student then check all the record data if match with entered ID is found.
- ✓ If match is found, do the following operations.
  - If the nth student’s GPA is less than the n-1th student’s GPA, swap nth student with n-1th student.
  - Continue this operation until the whole record is sorted.
  - If operation is finished without any problem, declare sorting is finished successfully.

8. Update student information either one by one or all at once.

- ✓ Prompt the user to enter which information he /she wants to update by using switch, it can be one by one or all information together.
- ✓ If user selects one by one updating of student information, Prompt user to enter new information corresponding to his/her option example: name or id.
- ✓ Place it on temporary variable, then copy the values on temporary variable to corresponding old variables holding previous values. If user selects updating all the information of student at once, Prompt user to enter new information corresponding to his/her option.
- ✓ Place all the student information on temporary variables, then copy the values on temporary variables to corresponding old variables holding previous values.

9. Traverse across all operations.

- ✓ Place all the menu options inside switch.
- ✓ Configure the switch to provide all the function calls on number labels given to them.
- ✓ Prompt user to enter his/her option based on integer number from keyboard input.
- ✓ Give corresponding functionality.
- ✓ Provide ability to quit the program when needed.