BAHIR DAR UNIVERSITY
Bahir Dar Institute of Technology
Faculty of Computing
Data Structures and Algorithms Linked List Project

Include the following C++ Functions in your Linked List Implementation.

1. Insert at the beginning of your linked list by reading from a notepad file.
2. Insert at the end of your linked list by accepting from keyboard.
3. Insert at the middle of your linked list after a given node in your linked list by accepting from keyboard.
4. Delete the first element in your Linked List.
5. Delete the last element in your Linked List.
6. Deleting functions that can delete a node with a given attribute: (i.e.,)
   6.1. Delete by Name
   6.2. Delete by IDNO
   6.3. Delete by Sex
   6.4. Delete by Department
   6.5. Delete by CGPA
   6.6. Delete by Age
   6.7. Delete by Faculty
   6.8. Delete by any other attributes identified in your structure definition.
7. Delete function that can delete the n^{th} node, given n. That means for example if a user enters number 5, it will delete the fifth node in the list, if a user enters number 9, it will delete the ninth node in the list, so in general if a user enters number n, it will delete the n^{th} node in the list.
8. Write a function that can search and find a student with MINIMUM CGPA, your function should assume that the linked list is not sorted in any order by any criteria (neither in ascending nor in descending order).
9. Write a function that can search and find a student with MAXIMUM CGPA, your function should assume that the linked list is not sorted in any order by any criteria (neither in ascending nor in descending order).
10. Write a function that can calculate AVERAGE Age of all students in the linked list.
11. Write a function that can calculate AVERAGE CGPA of all students in the linked list.
12. Write a function that can delete a student having MINIMUM Age, your function should assume that the linked list is not sorted in any order by any criteria (neither in ascending nor in descending order).
13. Write a function that can delete a student having MAXIMUM Age, your function should assume that the linked list is not sorted in any order by any criteria (neither in Ascending nor in descending order).
14. Add searching functions to search by each attributes identified in your structure definition: (i.e.,)
    14.1. Search by Name.
    14.2. Search by IDNO.
    14.3. Search by Sex.
    14.4. Search by Department.
    14.5. Search by CGPA.
    14.6. Search by Age.
    14.7. Search by Faculty.
    14.8. Or search by any other attributes identified in your structure definition.

(For all searching functions, the searched key value may be found or may not be found. If the searched key value is found, then display all information about those students, remember that there may be more than one student that satisfy the searching key criteria, that means for example, when you search by Name-Abebe- there may be 10 or 1000 or more students with name-Abebe-, so the program should display all information about all those 'Abebe's. If not found your program should print a message saying "NOT FOUND".)

15. Display all information of all Students in your Linked List. (for Double Linked List do for both Forward Display and Backward Display).
16. Write all information of all Students in your Linked List to a notepad file.
17. Write a function to count and tell the number of nodes (the number of Students) in your linked list.
18. Write functions to count number of occurrence of a certain value of an attribute in the list. (For example that count how many students are 19 years old, how many students are in Biology department, how many students have CGPA value of 3.2, how many students are female, how many students are in Computing faculty, how many students have name value of Abebe, and so on……).
    18.1. Count number of occurrence by Name
    18.2. Count number of occurrence by Sex
    18.3. Count number of occurrence by Department
    18.4. Count number of occurrence by CGPA
    18.5. Count number of occurrence by Age
    18.6. Count number of occurrence by Faculty
    18.7. Count number of occurrence by any other attributes identified in your structure
    18.8. definition.
19. Add update functions to update (modify) required attributes identified in your structure definition: (i.e.,)
    19.1. Update Name
    19.2. Update IDNO
    19.3. Update Sex
    19.4. Update Department
    19.5. Update CGPA
    19.6. Update Age
    19.7. Update Faculty
    19.8. Or update any other attributes identified in your structure definition.

    (for example, it should be possible to update age of a student from age value 20 to 22 if required using Update Age function).
20. Add sorting functions to sort by each attributes identified in your structure definition: (i.e.,)
    20.1. SortByName
    20.2. SortByIDNO
    20.3. SortBySex
    20.4. SortByDepartment
    20.5. SortByCGPA
    20.6. SortByAge
    20.7. SortByFaculty
    20.8. Or sort by any other attributes identified in your structure definition.
21. Write functions that display information about top students using attributes as criteria. (For example the first 10 students in their name alphabetical order, the first 7 students in their

IDNO order, the first 15 students in their CGPA order having maximum CGPA, the first 20 students in their Age order having minimum Age.

21.1. Top ten Name
21.2. Top seven IDNO
21.3. Top fifteen CGPA
21.4. Top twenty Age
21.5. Or Top student by any other attributes identified in your structure definition.

Instructions:

- ✓ Your code should be user-friendly (menu-based).
- ✓ Give special attention for Indentation. Indent your code (make readable). Indentations make it easy to find and fix errors in programming codes.
- ✓ Identify at least five attributes for your structure definition.
- ✓ For your variables choose appropriate variable-name, appropriate data type, and appropriate size.
- ✓ For your functions choose appropriate function-name, appropriate return-type, and appropriate parameter/s if required.
- ✓ When you accept data from keyboard, validate the data entered, for example Age cannot be negative integer. CGPA should be greater than 0 and less than 4.00. Apply validation when necessary.
- ✓ **Note:** During Insertion, make IDNO unique, that means it is not possible to register more than one student with the same IDNO.
- ✓ In your code include comments to make your code easily understandable.
- ✓ Write group members' name and IDNumber at the beginning of your code as a comment.
- ✓ Deliver the softcopy of your code before the deadline at the time of project group defense.
- ✓ Individual group members will defend the project.

For example for Student Registration System you can identify the following attributes:

Student:

- Name
- IDNO
- Sex
- Department
- CGPA
- Age
- Faculty and so on ………….