

---

# CST8234- C Language

## Assignment 2

Due Date - November 10, 2019

---

### Problem statment:

In this assignment, you will work on building almost a full linked list implementation. You will implement a list that stores students struct, which hold some of the student's information.

This assignment is intended to get you to practise using pointers, linked list, header files, and user inputs.

### Background Information:

This assignment is designed to help understand what linked lists are, how they work internally, and how to use them. You will implement the common functionality of linked lists and then use them to build a list of students.

Each student is represented only by their first and last name. Then, each student is linked to the next student in a system.

To encapsulate the data, which is the student in this case, we will create a node struct that will hold the student and a reference to the next student in the list.

The program will ask the user to enter each student information then proceed to build a linked list based on the user entry.

---

---

## Requirements:

Write a program that achieves the following requirements:

1. Implement all function defined in `Node.h` file in a separate `.c` file. (You should find the slides very helpful to complete the implementation).
2. The program should instruct the user on the actions they will perform while it is running all the time. (In other word, make sure you have meaning full and readable `printf` statement).
3. The program should have a structure that is called `student`, and type defined as `studnet_t`, with the following information:
  1. First name.
  2. Last name.
4. The program should have a structure that is called `Node`, and type defined as `node_t`, which has the following information:
  1. A student pointer, with represent the value of the node.
  2. A pointer to the next student in the list.
5. The program should have a pointer to a linked list called `head`, which is of type `node_t`.
6. The program should read the first 3 students from the user and add them to the list by adding to the beginning of the list.
7. Then, the program will read another 3 students' information but will add them to the end of the list.
8. The program then will delete the first 3 elements in the list.
9. Then the program will delete the last 3 elements in the list, emptying the list effectively.
10. The program will read 3 new student information from the user and add them to the end of the list.
11. Finally, the program will delete the second element in the list only, keeping the first and last in the list.

## Design Requirements:

1. You are given a header file, called `'Node.h'` which contain the prototypes of the functions you need to implement. **DO NOT** change any of the functions prototypes.
  2. Add in the 2 structs definitions in the header file so that you can use them everywhere.
  3. Create a new `.c` file also called `'Node'` to have the full implementation of the defined functions in it.
  4. In a separate file, declare the main function and only include the header file, not the `.c`.
-

- 
5. Implement the functionality described in the requirements section.
  6. **YOU MUST** build your code using all build flags mentioned in the slides as well as '-w' switch. Failing to do so will make you lose 1 point on the assignment.

## Sample Output:

```
First, you will enter 3 students names that will be added to the start of the list
Please enter first name for student 1: st1
Please enter last name for student 1: st1
Please enter first name for student 2: st2
Please enter last name for student 2: st2
Please enter first name for student 3: st3
Please enter last name for student 3: st3
st3 st3
st2 st2
st1 st1
Then, you will enter 3 students names that will be added to the end of the list
Please enter first name for student 1: st4
Please enter last name for student 1: st4
Please enter first name for student 2: st5
Please enter last name for student 2: st5
Please enter first name for student 3: st6
Please enter last name for student 3: st6
st3 st3
st2 st2
st1 st1
st4 st4
st5 st5
st6 st6
Then, you will remove the first 3 students in the list
st4 st4
st5 st5
st6 st6
Then, you will remove the last 3 students the list
By now, your list should be empty, so you will enter 3 more students
Please enter first name for student 1: st7
Please enter last name for student 1: st7
Please enter first name for student 2: st8
Please enter last name for student 2: st8
Please enter first name for student 3: st9
Please enter last name for student 3: st9
st7 st7
st8 st8
st9 st9
Finally, you will delete the second student in the list only
st7 st7
st9 st9
Program ended with exit code: 0
```

---

---

## Supporting files:

Along with this document, you will find a header file included in Brightspace. The file is named **node.h**. These file contains prototypes for functions that you will implement.

## Submission instructions:

1. No late submissions are accepted.
  2. You **MUST** work in a group of maximum 2 students to complete this assignment. Individual work is **NOT** accepted.
  3. You must submit the source code for the program you wrote.
  4. Add all your files under a folder call "lastName-firstName-Assign2", then Zip the folder and submit the zipped folder only.
  5. Make sure to submit all files required to compile and run the program on the instructor machine without any errors.
  6. **DON'T** submit any extra file. For example, the binary file (AKA object or output) files like .exe or .o.
  7. Add a Readme.txt file that contain each student name and student number.
  8. Brightspace is configured to keep the last submission only. Please make sure your last submission in the one you want to get marked.
  9. All submission must be done on the main Brightspace shell, 19F\_CST8234\_010\_ALL or 19F\_CST8234\_020\_ALL, not the lab section one.
-