# Data Structures and Algorithms Lab

**Lab 12** **Marks 07**

## Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

## Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

## What you must do

Implement the **StudentList** class which stores **Students** in unsorted order. Your class declarations should look like:

```
class Student
{
	friend class StudentList;

private:
	int id;			//id of a student.
	string name;		//name of a student.
	float cpga;		//cgpa of a student.
	Student* next;		//address of the next available object.

public:
	Student(int id, string name, float cgpa, Student *next);   //constructor
	void studentDetails();    //displays the student informaton
};

class StudentList
{
private:
	Student* head;			//start of the list
	Student* cursor;		//current item of the list

public:
	StudentList();			//constructor
	~StudentList();			//destructor
};
```

The **StudentList** class should also have the following public member functions:

### void insert (const Student& newStd)

Inserts **newStd** into a list. If the list is not empty, then inserts **newStd** after the **cursor**. Otherwise, inserts **newStd** as the first (and only) data item in the list. In either case, moves the cursor to **newStd**.

### void remove (int id)

Remove the particular **Student** based on the **id** and do not change the position of **cursor**. If the **cursor** pointing to the same object which is going to be deleted, then moves the **cursor** to the data item that followed the deleted data item. If the deleted data item was at the end of the list, then moves the **cursor** to the beginning of the list.

### void search (float cgpa) const

This function searches for **student(s)** based on his/her **cgpa** in the student list. It should dispaly all the information about the **student(s)** if found otherwise display an appropriate message.

### void replace (const Student& newStd)

Replace the data item with **newStd** based on the **id**. If no student exist with the **newStd**'s **id** then add the **newStd** at the end of the list. The **cursor** remains at **newStd**.

### bool isEmpty () const

Returns **true** if a list is empty. Otherwise, returns **false**.

### void gotoBeginning ()

Moves the cursor to the beginning of the list

### void gotoEnd ()

Moves the cursor to the end of the list.

### bool gotoNext ()

If the cursor is not at the end of the list, then moves the cursor to mark the next data item in the list and returns **true**, otherwise returns **false**

### bool gotoPrior ()

If the cursor is not at the beginning of the list, then moves the cursor to mark the preceding data item in the list and returns **true**, otherwise returns **false**

### Student getCursor ()

Returns a copy of the data item marked by the cursor.

### void showStructure () const

Outputs the data items in a list. If the list is empty, outputs “Empty list”.

In the **main** function, your program should take the data of students from a text file **input.txt** and store the info of each student into the **student list**. The file is in the following format: **id**; line break, **name**; line break, **cgpa** and then a blank line followed by the data of next student, exactly in the same order as described above.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺