2E3 Lab 4

1. You will create a class to create student objects that can be connected via pointers to form a linked list:
   - Define a class StudentNode with data members: `int ID; string firstName; double finalGrade; StudentNode* nextS;`
   - All data members should be private so you will need getter and setter methods accordingly.
   - Keep track of the number of student nodes created with a static variable numStudents.
   - The StudentNode constructor should read in and set the name and grade, set the ID based on the number of students then increment numStudents, and set the nextS pointer to NULL;
   - Provide a second constructor that reads in the id, name and grade, and sets the id to the first value and not the numStudents value.

2. Implement a class called StudentLinkedList which will links together StudentNodes:
   - Define and implement a constructor that sets the headP and tailP to NULL.
   - The destructor should delete all the nodes at the end.
   - Implement the following methods:
   - `void printAllStudents() //loop through from head to tail, printing the data`
   - `StudentNode* findByName(string) //loop through until the name is found, then return that node`
   - `StudentNode* findByID(int); //ditto for ID`
   - `void addStudent(string, double); //add a new node`
   - `bool removeStudent(string);` //find the student with the given name and delete him/her
   - `bool removeStudent(int);` //find the student with the given ID and delete him/her
   - `bool removeStudent(StudentNode*);` //delete the student pointed to
   - `bool insertStudentAfter(StudentNode*, string);` //insert the student node after the student named

3. In your main program (Lab4.cpp):
   - First test your StudentNode class by writing a main program to dynamically allocate a node: `StudentNode* s0=new StudentNode(100, “John”, 45);` then call the methods to read his marks and print out his final grade.
   - create a StudentLinkedList object, sll0
   - Let the user input the names and grades for 10 students and add a node to the linked list for each one.
   - Print out all the data in the nodes.
   - Pick one of the names you entered (or let the user pick one), and find and print that student’s ID and grade, then remove the student. Do the same for the ID of another student.
   - Similarly, pick any remaining student’s name (not in the head or the tail node), and insert the node s0 after it.

- Print out the final list.

You have <u>two weeks</u> to complete this program, and the marks will be assigned as follows:

This week – Week6:
0 = did not attend lab
1 = attended the lab session

Max = 1 point

Next week – Week7:
Reading week – no lab

In 2 weeks – Week 8:

0 = did not attend lab
1 = attended the second (marking) lab session but have very little working code or the code does not compile. **Your code MUST compile in order to get a mark higher than 1**
2 = correct definition and implementation of the StudentNode class
4 = correct definition and implementation of the StudentLinkedList class
2 = correct implementation of the main program

Max = 9 points

<u>Overall maximum for two labs together= 10 points</u>

The program needs to be DEMONSTRATED for marks to one of the demonstrators before the end of your lab session in two weeks (Week 8) AS WELL AS SUBMITTED ONLINE through Blackboard.