CS2613: Programming Languages Laboratory
Python: Question 3

Overview:

The following question must be completed in Python 3. Submissions must be made to GitHub. Late submissions will not be accepted. All online resources are available including official documentation, forums, videos, etc. If you are struggling with a concept and online research is not helping, then you should ask the course professor.

Learning Goals:

- Expressions, Statements, Variables
- Math
- Conditionals
- Looping
- Searching and Sorting
- Object Oriented Programming
- Data Structure Creation

Question: Student Class List Organizer

Write a program that will allow a user to input the name of a course and the maximum number of seats. Then, the user should be able to interact with a textual user interface to add students, remove students, and print information about the class list according to the specifications described below.

First, all students (name + student id) must be stored as objects in a **doubly linked list of nodes**. You are required to program the doubly linked list class and any other classes to support it. You may **not** import or use any other data structures to create the doubly linked list (this includes **no arrays or lists**).

Required methods for the doubly linked list:

1. **num_students** – returns the current number of students registered in the course.

2. **add_student** – takes a Student object and adds it to the course in order of student ID.

3. **remove_student** – takes an integer (student ID) and removes the first occurrence of that student id from the course while maintaining order.

4. **print_ascend** – return a string containing all students in the class (name + id) in ascending order of id. String should be readable if printed out immediately after return.

5. **print_descend** – return a string containing all students in the class (name + id) in descending order of id. String should be readable if printed out immediately after return.

Textual User Interface Requirements:

#	Description
1	Asks for a new student's name and ID. Adds new student to the class list. Students should be sorted by student ID. Cannot add to a full class. Should print a success or failure message.
2	Asks for a student's ID. Removes that student from the list. Cannot remove a student if they are not on the list. Order should be maintained.
3	Print the number of students in the class
4	Print the class list in ascending order by ID. Must move forward through the doubly linked list.
5	Print the class list in descending order by ID. Must move backward through the doubly linked list.
0	Quit the program
Any Other Number	Print the instructions again. Instructions do not need to be re-printed after each selection 0-5, only when the user requests them.

Example I/O:

```
Enter class name and max number of students separated by the ENTER key
CS2613
5
Please make a selection for the course CS2613
      1:    Add Student
      2:    Remove Student
      3:    Print Number of Students
      4:    Class List in Ascending Order
      5:    Class List in Descending Order
      0:    Exit Program
1
To add: Input student name and student number (separated by ENTER key)
Student C
3
Added successfully
1
To add: Input student name and student number (separated by ENTER key)
Student A
1
Added successfully
1
To add: Input student name and student number (separated by ENTER key)
Student E
5
Added successfully
3
Number of students in the class: 3
4
Student A:        1
Student C:        3
Student E:        5

5
Student E:        5
Student C:        3
Student A:        1

2
To remove: Input student's unique id
3
Student successfully removed
4
Student A:        1
Student E:        5

9
Please make a selection for the course CS2613
      1:    Add Student
      2:    Remove Student
      3:    Print Number of Students
      4:    Class List in Ascending Order
      5:    Class List in Descending Order
      0:    Exit Program
0
```