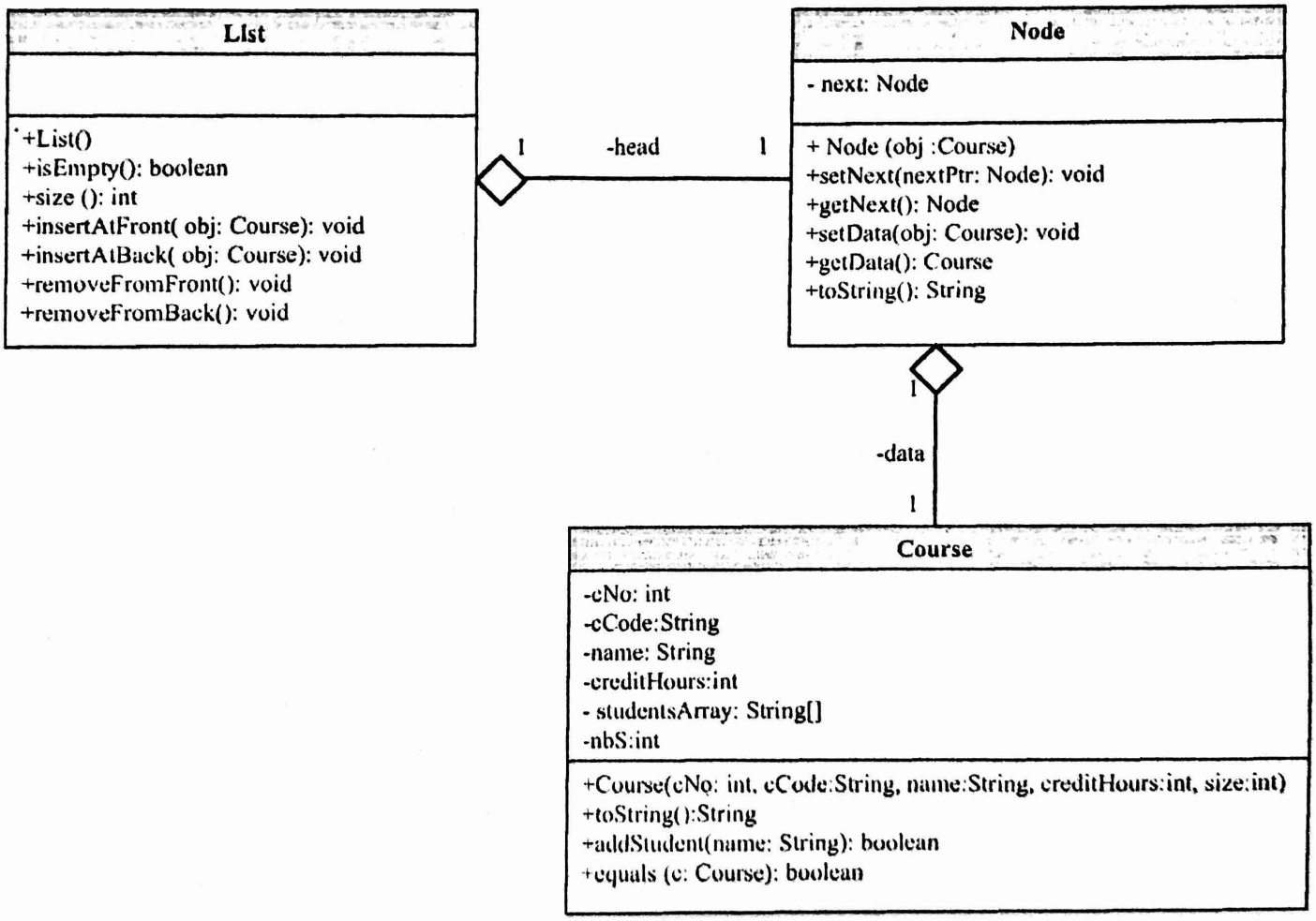
IMPORTANT:

- ✓ Create folder in the desktop with your name (FirstName_LastName_ID) and save all your work on it.
- ✓ Compress (Zip) this folder and submit it through LMS (Lab-> FinalLab).
- ✓ FinalLab duration is 3 hours.

1. The classes List and Node have already been implemented for you, you need to use them properly.

2. Define a new checked exception of type InvalidCourseNoException.

3. Implement Class Course:



Attributes:

- **cNo: int** → The number of the course such as *113*
- **cCode: String** → the code of the course such as *CSC*
- **name: String** → The name of the course such as *Java Programming 2*
- **creditHours:int** →The total number of hours per week for this course.
- **studentsArray: String[]** → An array containing names of students who are enrolling in the course.
- **nbS: int** →An integer referring to the first empty entry in the studentsArray.

Methods:

- **Course(cNo: int, cCode:String, name:String, creditHours:int, size:int)**→Constructor to initialize the course information.

  Hint: size is the length of the array.
- An **InvalidCourseNoException** should be generated and caught in the same environment if the cNo is less than 111 or if it is more than 499 allowing the user to re-enter the wrong input.
- **toString(): String** → Returns the course's information (including all students) in a string.
- **addStudent(name: String): Boolean** →Adds the received name to studentsArray. Return true if the name is added successfully, false otherwise.
- **equals (c: Course): boolean** →Compares this course to the received course. The result is true if and only if **c** is not null and **c** has the same **cNO** and **cCode** as the current course.

4. Create a new Application class with main method to perform the following:

a. Using the class List, create a stack of Course named **cList**.

Hint: In Stack, the order is Last In First Out (LIFO).

b. **Add three Courses** to the list. Ask the user to enter their information. You need to use the appropriate methods.

c. For each Course in cList do the following:

i. Save Course's information in a text file named **"All Courses.txt"**.

ii. Add a new student named "Amal" to the course.

iii. Print the course's information.

iv. Save the Course in an object file named **"Course.data"**.

d. Create a Course object name **course1** with the following information:

cNo	113
cCode	CSC
name	Java Programming 2
creditHours	3
studentsArray	{"Sara", "Nora", "Maha", "Arwa"}
nbS	4

e. Display a meaningful message indicating if the course **course1** exists in cList or not.