UML Diagram:

<<interface>>
java.io.Serializable

//creates .ser files and stores
//serialized data

<<interface>>
java.lang.Comparable

compareTo(other : Object) : int

<>   **User**
username: String
password: String
firstName: String
lastName: String
fullName: String
User()
User(username: String, password: String)
User(username: String, password: String, firstName: String, lastName: String)
permission(): boolean
getUsername(): String
getPassword(): String
getFirstName(): String
getLastName(): String
getFullName(): String
setUsername(String s): void
setPassword(String s): void
setFirstName(String s): void
setLastName(String s): void

`*` `1`

`1` `*`

Course
courseName: String
courseID: String
maxStudents: int
currentStudents: int
fullName: String
nameList: List<String>
courseInstructor: String
CourseSectionNumber: String
CourseLocation: String
Course()
Course(courseName: String, CourseID: String, maxStudents: int, currentStudents: int, nameList: String, courseInstructor: String, courseSectionNumber: String, courseLocation: String)
addStudent(fullName: String, username: String)
addStudent(fullName: String) //overloading
removeStudent(fullName:String)
isNotFull(): boolean
//all getters and setters...

Student
courseList: List<Course>
super(); ------- //(gains other constructors too)
getCourseList(): List<Course>
printCourses(): void
addCourse(): void
removeCourse(): void

Admin
permission():boolean //overriding

<<interface>> **StudentMethods**
addCourse(): void
removeCourse(): void
getCourseList(): List<Course>

<<interface>> **AdminMethods**
//getters and setters

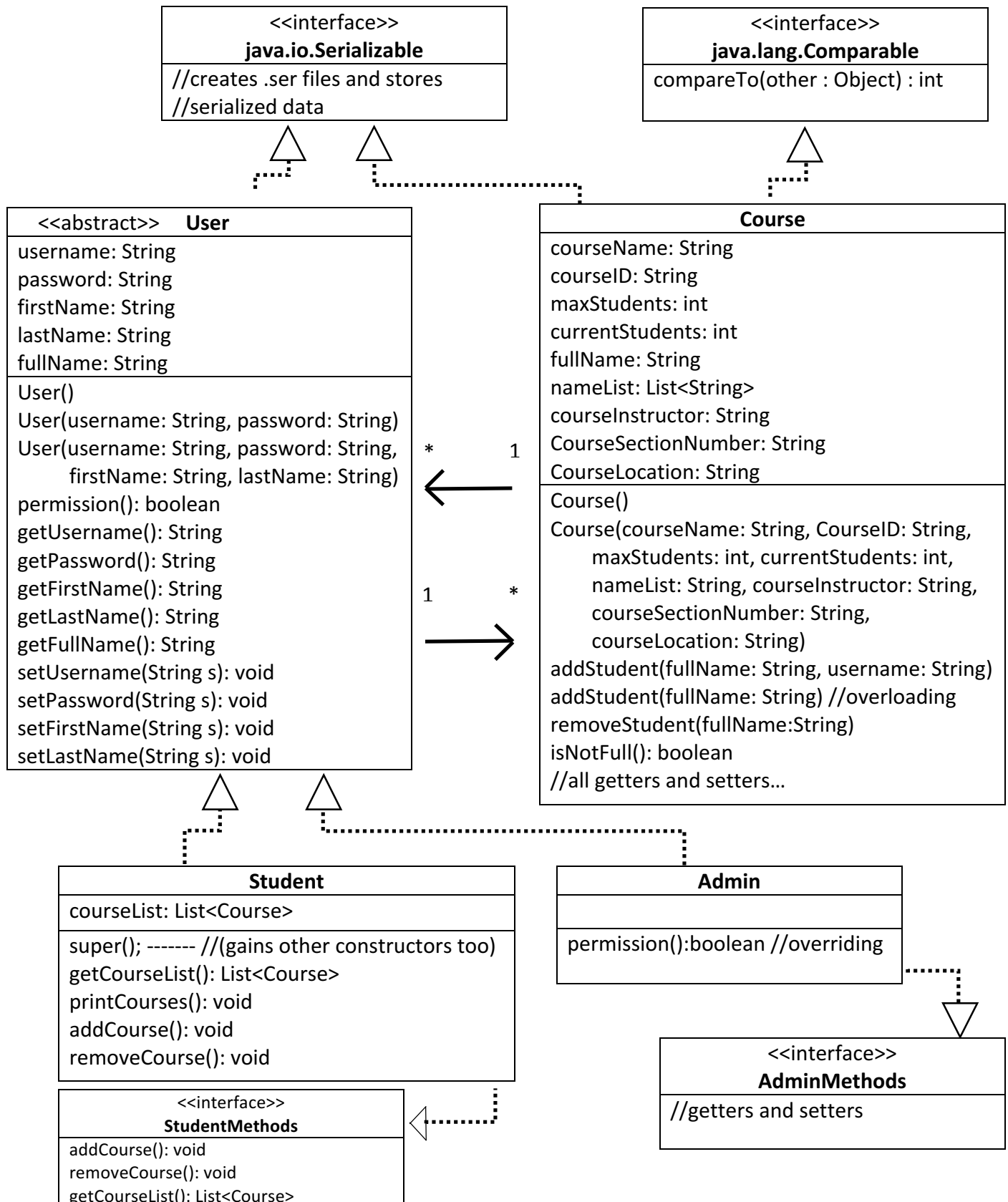**Workflow:**

(main method –

both ArrayLists are initialized (courseList and studentList)

if .ser file does not exist, import from CSV. Otherwise, de-serialize)

- Display asks for login – username and password
    - Determines Admin or Student
- If Admin:
    - Go to MainMenuAdmin
    - **10 input options (switch function):**

5 course management:

- createCourse();
    - creates a new course object based on inputted parameters
    - added to courseList;
- deleteCourse();
    - deletes a course object in courseList;
- editCourse();
    - 6 options on how to edit a course object
    - Change:
        - maxStudents; courseInstructor; courseSectionNumber; courseLocation; Add/Remove Students from Course
- displayCourseInfo();
    - displays parameters for given course, based on courseID
- createStudent();
    - creates new Student object based on inputted parameters
    - added to studentList;

5 reports:

- View All Courses
- View All Full Courses
    - Writes to FullCourseList.txt
- View the Student list for a specific Course
- View the Course list for a specific Student
- Sort the Course list by currentNumber of Students
    - (using Comparable interface)

AFTER FUNCTION IS COMPLETED, GOES BACK TO MainMenuADMIN.

- If Student:
    - Go to MainMenuStudent
    5 input options (switch function, again)

        - View All Courses
        - View all Courses that are *not* full
        - registerStudent(); -- similar to Add Students to Course from editCourse();
        - withdrawStudent(); -- similar to Remove Students from Course
        - View All Courses that current student is registered in

AFTER FUNCTION IS COMPLETED, GOES BACK TO MainMenuSTUDENT