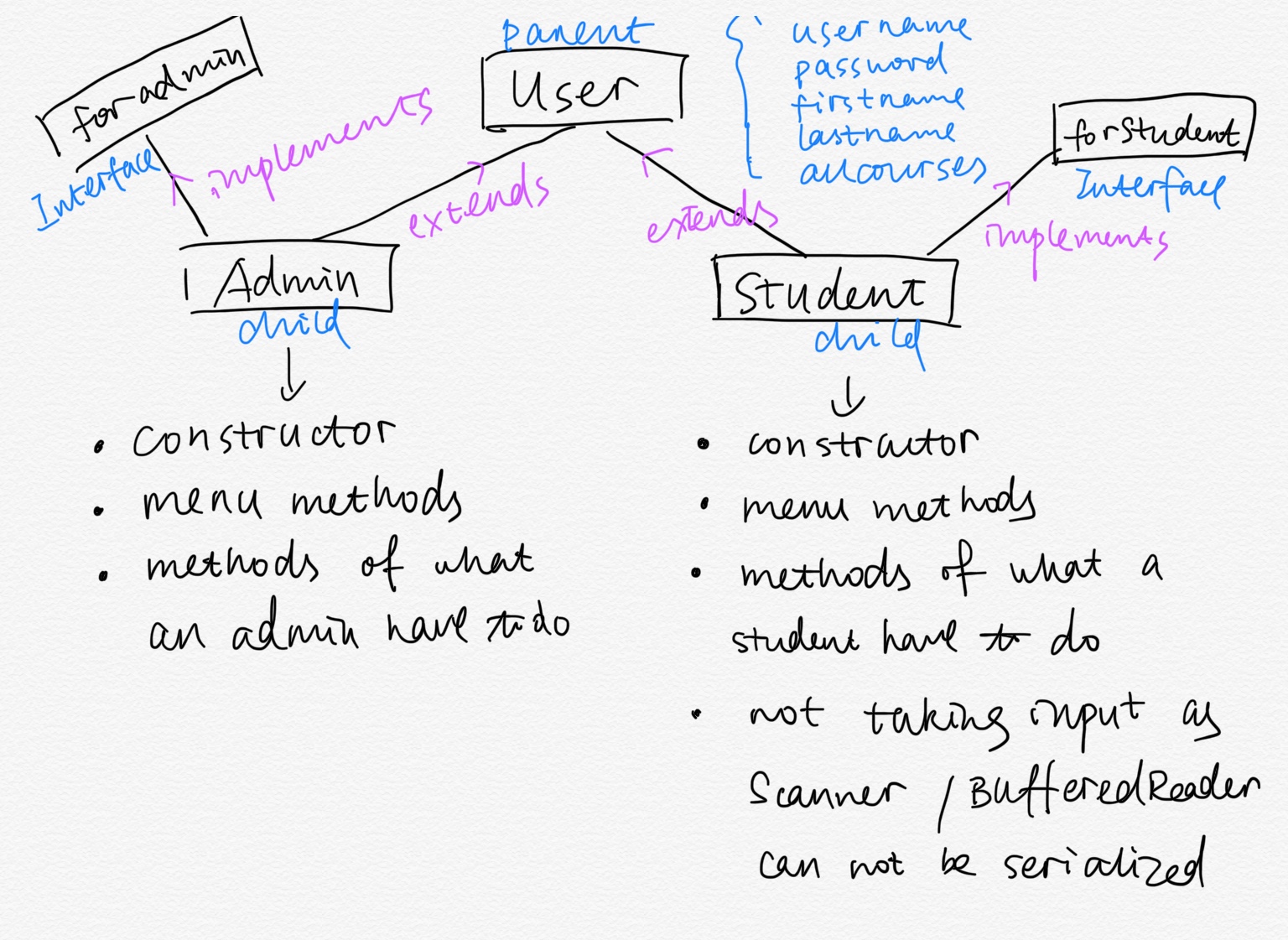
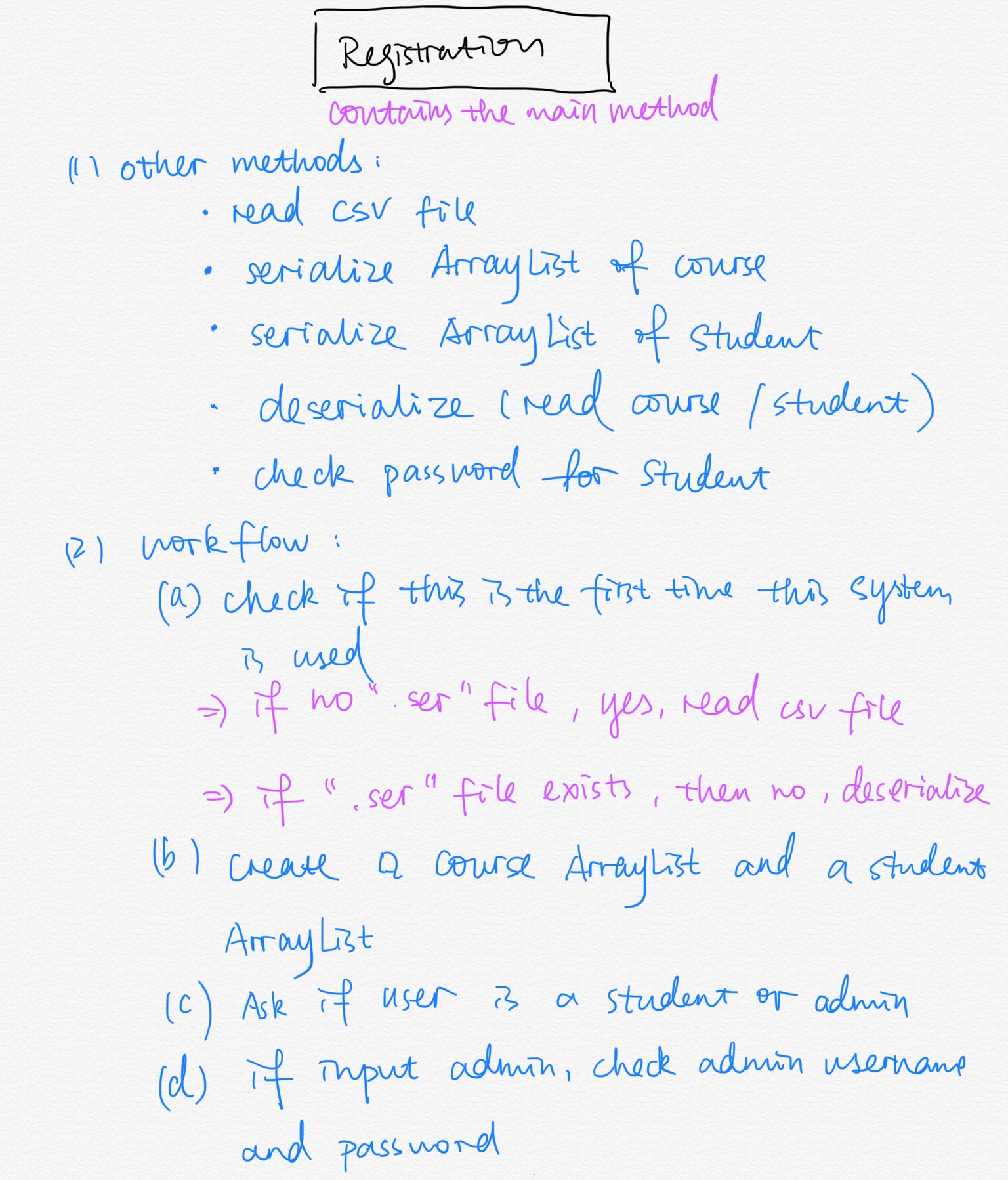
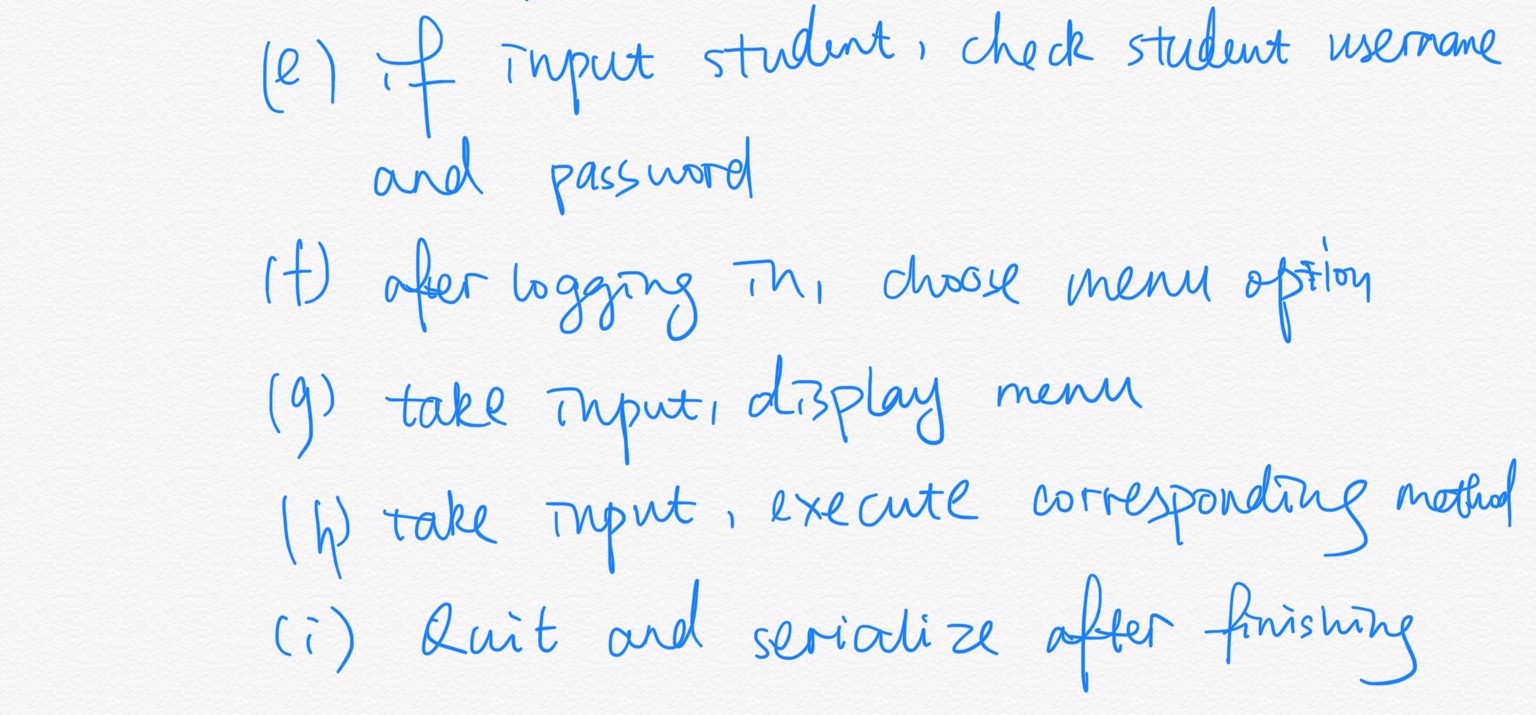
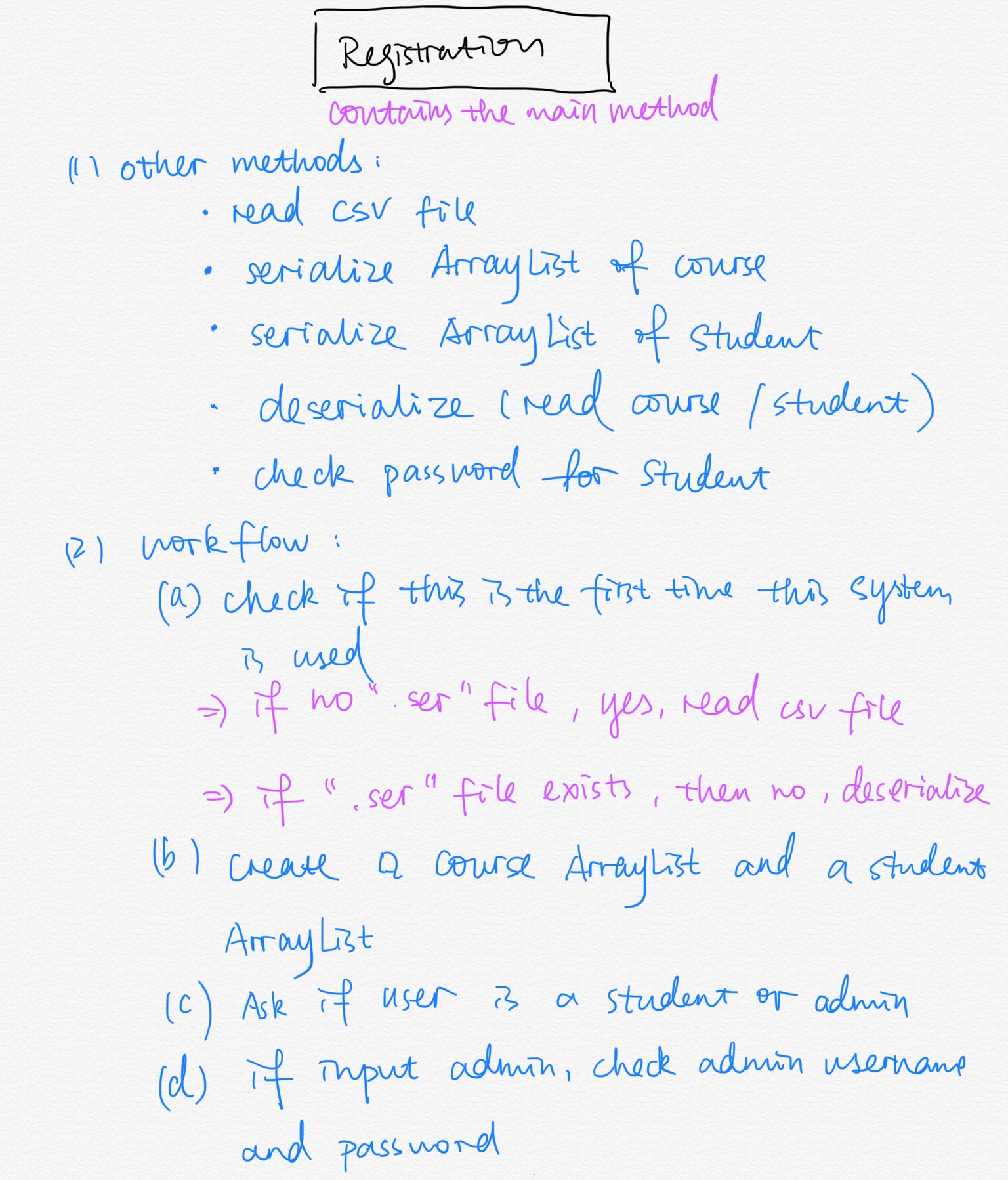
Work Flow





Concepts

▪ Method overloading

Didn’t really use it, but it can be

public void a (int m){  
 System.out.println(m);

}

public void a (double d){  
 System.out.println(d);

}

▪ Method overriding :

1. In class admin: @Override

**public** **void** editCourse() {}

1. In class student: @Override

**public void** viewAllCourse() {}

▪ Abstract Class

Interfaces forAdmin and forStudent are abstract. Methods are initiated but not

Implemented: **public** **void** createCourse();

**public** **void** deleteCourse();

▪ Inheritance

Admin and Students are inherited from User class, some data fields and methods from

User class can be used in child class directly.

▪ Polymorphism

Admin and student, both extends user class and have a same method:

public void menuCM() {}

this method will be used diffenrently depents on Admin.menuCM();

or Student.menuCM();

▪ Encapsulation

In registration class, we have code: **if** (choice == 1) {

a.viewAllCourse();

a.menuRe();

That is, we hided the code in another class, which we hided th complexity of the coding,

and let users see the call only; also, it makes the main method easier and more structured to use

▪ The concept of ADT (Abstract Data Types)

We use arraylist and list ADT: *allcourses*.remove(i);