Assignment 3 Part 2 - Design Violations

- Student header file:
    - Has bad cohesion, and violating the Interface Segregation Principle (ISP)
    - All course grade related methods should be in transcript, student header file Nameshould strictly contain information regarding the student's Name,Number and Email
    - Operator comparison on students seems over designed, unnecessary
        - Wrong syntax too as it is comparing a Student object to a Course object

- RegistrationImpl header file
    - The whole header file seems unnecessary as it violates the Don't Repeat Yourself (DRY) Principle
    - bloated code and decreased efficiency

- Conflict between Registration and Course classes
    - Content Coupling occurs between Registration and Course classes, Registration will change the course class's status from OPEN to WAIT_LIST to CLOSED, also occurs for course dropping changing the course status

- Coincedental Cohesion in Course
    - The entire Course object has bad cohesion which makes it hard to follow, which will result with low maintainability and decreased efficiency
    - Course object is first normal form and can be upgraded to a higher normal form

- Registration header file
    - When registering for a class, it access prerequisites and skipping access through course, violates Law Of Demeter (LoD), it goes registration->prerequisites instead of registration - > course - > prerequisites

- Information hiding can be applied into CourseCatalog
    - The attribute courseList with the type set<Course> can be privatized as it not needed