**OOP Concepts with Examples**

1. **Method Overloading**

In the User class:

Method 1: +isInCourseList(courseID: String): int

Method 2: +isInCourseList(name:String, courseID:String, sectionNum: int) : int

These two methods have the same name but different parameters.

1. **Method Overriding**

Example 1:

Course class overrides the compareTo(Object o) method in Comparable<E> interface

Example 2:

Student/Admin class overrides all the methods in Student/Admin interface

1. **Abstract Class**

The User class is an abstract class with abstract methods setUsername() and setPassword();

1. **Inheritance**

The Admin and Student classes inherit from User class.

1. **Polymorphism**

In the Main class, there is a method: +userAction(action: String, u:User) which the instances of both the Admin class and the Student class can use. This is because the Admin class and Student class both inherit from the User class. An object of a subtype can be used wherever its supertype value is required. This feature is known as polymorphism

1. **Encapsulation**

Almost all the data fields in Admin/Student/User/Course are private but I create several getters and setters for them.

1. **The concept of ADT (Abstract Data Type)**

In C++, ADT are realized by struct, union, and class. In C, they can be realized by struct. In java, we can do that using class. In this java project, I created 8 classes, such as Course, Student, Admin, etc. In these classes, the User class is an abstract class.