Assignment 4

Requirements:

- Create a Java project named yourStudentId_OOP_HW4
- Read instructions and create classes needed. You are supposed to add 3 classes (2 required + 1 Tester) to the project.
- Your code must be properly formatted with sensible variable names! Refer to the text for code format examples.
- The instruction for Tester and output are for your reference.
- Make sure your classes correctly implement the public interfaces.

The following diagram describes two class you need to implement.

Student
studentID: int studentName: String enrolledCourses: ArrayList<Course> currentCredits: int maxCredits: int
setStudentName(String): void setCurrentCredits(int): void setMaxCredits(int): void getStudentID(): int getStudentName(): String getEnrolledCourses(): ArrayList<Course> getCurrentCredits(): int getMaxCredits(): int getCourse(int): Course enroll(Course): void drop(int): void drop(Course): void getInfo(): String

Course
courseID: int courseName: String credits: int capacity: int enrolled: int
setCapacity(int): void setEnrolled(int): void getCourseID(): int getCourseName(): String getCredits(): int getCapacity(): int getEnrolled(): int getInfo(): String

Figure 1

1. Create Student class

Student	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	studentID The student ID.
String	studentName The student's name
ArrayList<Course>	enrolledCourses An ArrayList that holds all courses have been selected
int	currentCredits current credits

int	maxCredits credits limit
Constructor	
Student(int studentID, String name) Enable to construct a student object with given *student id, name*, set *currentCredits* as 0, set *maxCredits* as 25 and initialize *enrolledCourse*. **Student(int studentID, String name, int maxCredits)** Enable to construct a student object with given *student id, name, maxCredits*, set *currentCredits* 0 and initialize *enrolledCourse*.	
Instance methods	
-	For setter and getter please refer to Figure 1.
Course	getCourse(int id) Find the Course object in *enrolledCourses* by the *courseID*. If found, returns the Course object. Otherwise, returns null.
void	enroll(Course course) This method is for student to register a course. 1. Check whether the capacity of the course is available. If not, print *"courseID is full"* and go to step 6. 2. Check if the *currentCredits* after adding the course is less than *maxCredits*. If not, print *"studentID cannot enroll any course"* and go to step 6. 3. Add the course to *enrolledCourse* 4. Adjust enroll number of the course 5. Adjust *currentCredits* of the student 6. Terminate the program
void	drop(int courseID) Use parameter *courseID* to check if the course is in *enrolledCourses* or not. If is, remove it from *enrolledCourses*(Don't forget to revise the current number of student of this class and adjust the current credits of this student);If isn't, show the information at console. (See output below)
void	drop(Course course) Use a Course object to check if the course is in *enrolledCourses* or not. If is, remove it from *enrolledCourses*(Don't forget to revise the current number of student of this class and adjust the current credits of this student);If isn't, show the information at console. (See output below)
String	getInfo() Return a String description for the student information and the course student has chosen. (See output below)

2. Create **Course** class

Course	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	courseID The course number of this course.
String	courseName The course name of this course.
int	credits The credits of the course.
int	capacity The maximum number of students in this course
int	enrolled The current number of students in this course.
Constructor	
Course(int id, String name, int credits, int capacity) Enable to constructs a Course object with given *student id, name, credits, capacity* and set *enrolled* as 0.	
Instance methods	
-	For setter and getter please refer to Figure 1
String	getInfo() Return a String description for the *courseID, courseName, credits, enrolled and capacity* of the course. (See output below)

Tester

```
public class Tester {
        public static void main(String[] args) {

                Student stu1 = new Student(109306100, "Allen", 15);
                Student stu2 = new Student(109306101, "Bob");
                Course c1 = new Course(306001, "OOPI-1", 2, 10);
                Course c2 = new Course(306011, "OOPI-2", 2, 20);

                stu1.enroll(c1);
                stu2.enroll(c2);
                stu2.enroll(c1);
                stu1.enroll(c2);
                stu1.drop(306001);
                stu2.drop(c2);
                System.out.println(stu1.getInfo());
                System.out.println("---------");
```

```
                System.out.println(stu2.getInfo());
                System.out.println("---------");
                System.out.println(c1.getInfo());
                System.out.println(c2.getInfo());
        }
}
```

Output

```
Student ID: 109306100
Name: Allen
Credits: 2/15
Course list:
306011 OOPI-2 2

---------
Student ID: 109306101
Name: Bob
Credits: 2/25
Course list:
306001 OOPI-1 2

---------
306001 OOPI-1 2 1/10
306011 OOPI-2 2 1/20
```

Submission: Submit your project as **".zip file"** via Moodle. No other submissions will be graded.

Reminder: Please zip the whole project

Deadline: 2020/11/30 (for Mon56) or 2020/12/01 (for Tue23)