Assignment 5

Requirements:

- Create a Java project named yourStudentId_OOP_HW5
- Read instructions and create classes needed. You are supposed to add 5 classes (*Person, Student, Instructor, DoubleMajor, and Tester*) to the project.
- All instance variables are private. Please use public interfaces to access private variables.

Following figure shows the inheritance, attributes and methods for each class. For setter and getter please refer to Figure 1.

Description: The HW mainly simulates the relationship between instructors and students in the campus. Both the instructor and the student are individuals (Person). Therefore, the question can be completed by using the "Inheritance" concept taught in these few weeks. In addition, among the students, there will be students with double majors (the question only consider student with / without double majors). The student with double majors also has *major2* instance variable and corresponding methods.

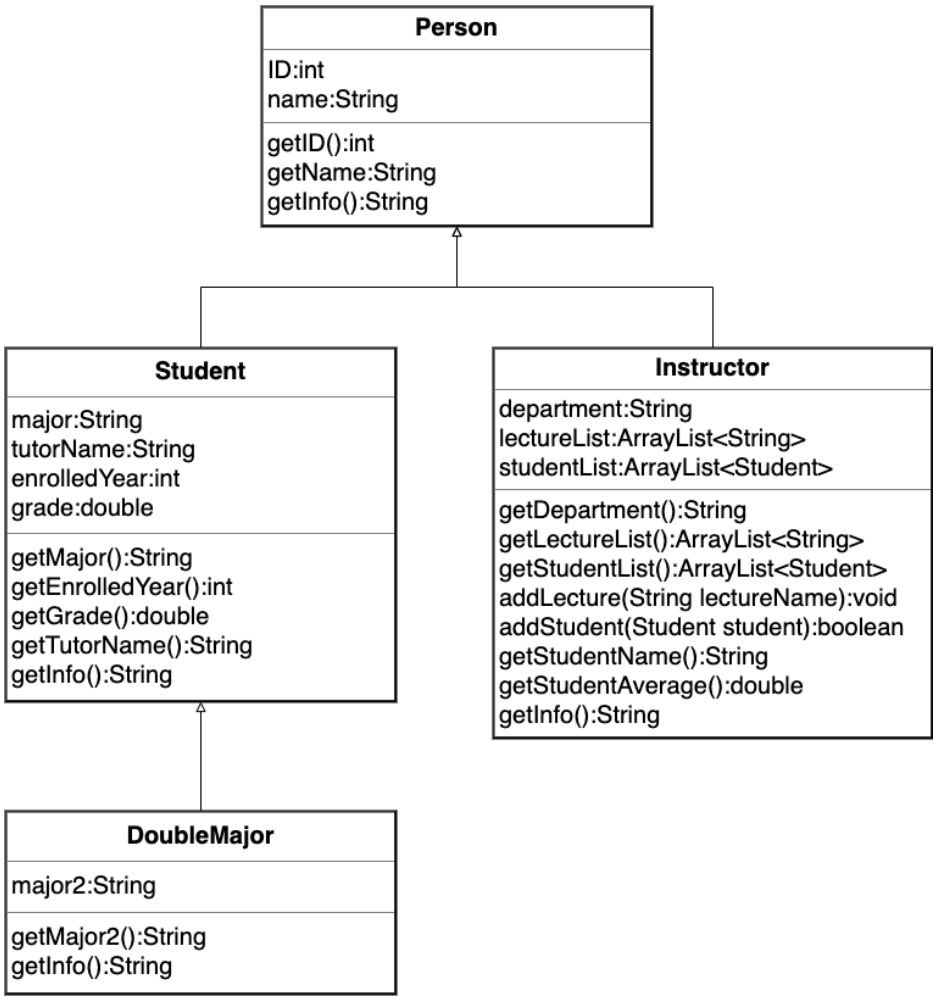

Figure 1.

1. Create **Person** class

Person	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	ID The person's ID.
String	name The person's name.
Constructor	
Person(int ID, String name) Enable to instantiate a Person object with given ID and name.	
Instance methods	
-	2 getter for 2 attributes (getID(), getName()).
String	getInfo() a. Returns a String description of the person's information. **Example:** Person[ID=109306501, name=Alex]

2. Create **Student** class

Student	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	major The major of the student.
String	tutorName The name of the student's tutor.
int	enrolledYear The year that the student enrolled.
double	grade The final grade that the student gets.
Constructor	
Student (int ID, String name, String major, int enrolledYear, String tutorName, double grade) Enable to instantiate a Student object with given ID, name, major, enrolledYear and initialize gradeList.	
Instance methods	
-	4 getters for 4 attributes (getMajor(), getEnrolledYear(), getGrade(), getTutorName()).
String	getInfo() a. Returns a String description of the student. **Example:** Student[ID=1093060502, name=Amy, major=MIS, enrolledYear=107, grade=95.00]

3. Create **Instructor** class

Instructor	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	department The department of the instructor.
ArrayList<String>	lectureList An ArrayList that contains the lectures of the Instructor.
ArrayList<Student>	studentList An ArrayList that contains the students of the Instructor.
Constructor	
public Instructor(int ID, String name, String department) Enable to instantiate an Instructor object with given ID, name, department and initialize lectureList and studentList.	
Instance methods	
-	3 getters for 3 attributes (getDepartment(), getLectureList(),getStudentList()).
void	addLecture(String lectureName) a. Add incoming lecture into the lectureList.
boolean	addStudent(Student student) a. Check the student's tutor. If the student's tutor name is the same as this instructor, add this student into the studentList and return "true", otherwise return "false".
String	getStudentName() a. Return all students' name in the studentList. **Example:** Amy, Joy
double	getStudentAverage() a. Calculate and return the average score of students that taught by the tutor.
String	getInfo() a. Return a String description of the instructor. **Example:** Instructor[ID=111306100,name=Jennifer,department=MIS,lectureList=OOPI, OOPII, studentList=Amy, Joy]

4. Create **DoubleMajor** class

DoubleMajor	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	major2 The second major of the student.

Constructor	
DoubleMajor(int ID, String name, String major, int enrolledYear, String tutorName, double grade, String major2Name) Enable to instantiate a DoubleMajor object with given ID, name, major, enrolledYear, tutorName, grade, and major2Name.	
Instance methods	
-	1 getter for 1 attribute (getMajor2()).
String	getInfo() a. Return a String description of the double major student. **Example:** DoubleMajor[ID=1093060504, name=David, major=MIS, major2=CS, enrolledYear=106, grade=89.00]

Reference code

```
public class Tester {

	public static void main(String[]args) {

		Person person = new Person(109306501, "Alex");

		Instructor tutor1 = new Instructor(111306100, "Jennifer", "MIS");
		Instructor tutor2 = new Instructor(111306101, "Bob", "MIS");
		tutor1.addLecture("OOPI");
		tutor1.addLecture("OOPII");
		tutor2.addLecture("OOPI");

		Student student1 = new Student(1093060502, "Amy", "MIS", 107,
					tutor1.getName(),95);
		Student student2 = new Student(1093060503, "Joy", "Acc", 108,
					tutor1.getName(),88);
		DoubleMajor student3 = new DoubleMajor(1093060504, "David", "MIS", 106,
					tutor2.getName(), 89, "CS");

		System.out.println(tutor1.addStudent(student1));
		System.out.println(tutor1.addStudent(student2));
		System.out.println(tutor1.addStudent(student3));

		System.out.println(person.getInfo());
		System.out.println(tutor1.getInfo());
		System.out.println(student1.getInfo());
		System.out.println(student2.getInfo());
		System.out.println(student3.getInfo());
		System.out.println("----------------------");
		System.out.println(tutor1.getStudentName());
		System.out.println(tutor1.getStudentAverage());
	}
}
```

Sample output

```
true
true
false
Person[ID=109306501, name=Alex]
Instructor[ID=111306100, name=Jennifer, department=MIS, lectureList=OOPI, OOPII, studentList=Amy,
Joy]
Student[ID=1093060502, name=Amy, major=MIS, enrolledYear=107, grade=95.00]
Student[ID=1093060503, name=Joy, major=Acc, enrolledYear=108, grade=88.00]
DoubleMajor[ID=1093060504, name=David, major=MIS, major2=CS, enrolledYear=106, grade=89.00]
-----------------------
Amy, Joy
91.5
```

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

Reminder: Please zip the whole project.

Deadline: 2020/12/28 (for Mon56) or 2020/12/29 (for Tue23)