Assignment 3

Requirements:

- Create a Java project named yourStudentId_OOP_HW3
- 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 outputs is your reference.
- Make sure your classes correctly implemented with the public interfaces.
- Note that all instance variables are private. Please use public interfaces to access private variables.
- The following diagram describes two class you need to implement.

Student
int studentID String name String department int[5] grades int gradesIndex
void setStudentID(int) void setName(String) void setDepartment(String) int getStudentID( ) String getName( ) String getDepartment( ) int[] getGrades( ) int getGradesIndex( ) int getGrade(int) void addGrade(int) void updategrade(int, int) String info( )

Grading
int passMark
void setPassMark(int) int getPassMark( ) String toLetterGrade(int) double calculateAvg(int[ ]) String summarizeGrade(int [ ])

1. Create **Student** class

Student	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	studentID The student ID.
String	name The student’s name
String	department The department the student belongs to.
int[ ]	grades

	An array that can store 5 grades.
int	gradesIndex The initial value is 0. This variable is used as a counter for grades.
Constructor	
Student(int studentID, String name, String department) **Constructs a student object with given student id, name, and department, and an empty array of grades.**	
Instance methods	
-	3 setter for 3 attributes. 5 getter for 5 attributes.
int	getGrade(int idx) Gets the value in grades by specific index.
void	addGrade(int grade) If gradesIndex is in valid range, add a new grade to grades at gradesIndex and gradesIndex + 1. If the index is out of bound, that is, gradesIndex is 5 or greater, print an error message "**Array index out of bounds**".
void	updateGrade(int idx, int grade) Updates the value in grades at given index. (Suppose idx is in valid range)
String	info() Returns a **formatted String** that describe the information about the student. (See sample output.)

2. Create **Grading** class

Grading	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	passMark The pass marks. (For example, the pass mark for undergraduate school is 60.)
Constructor	
Grading(int passMark) **Constructs a grading object with given passMark.**	
Instance methods	
-	one getter for an attribute. one setter for an attribute.
String	toLetterGrade(int score) Converts the grade to the corresponding letter grade and returns it (see table 1 for grade reference). Table 1 Grade reference

Score Range	Letter Grade
100 ~ 80	A
70 ~ 79	B
60 ~ 69	C
50 ~ 59	D
1 ~ 49	E
0	X

double	calculateAvg(int[] grades) Calculates the average of the input array and returns the avg. score. **(To make it simple, don't count 0)**
String	summarizeGrade(int[] grades) Returns a string that describe the average score, and pass/failed count **(Don't count 0)** of the input parameter. (See sample output.) Must call calculateAvg(…).

Tester

```
public class Tester {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Student stu1 = new Student(...);
		Grading grading1 = new Grading(60);
		stu1.addGrade(100);
		stu1.addGrade(70);
		stu1.addGrade(50);
		stu1.addGrade(67);
		stu1.addGrade(98);
		stu1.addGrade(90);

		System.out.println("------info()");
		System.out.println(...);
		System.out.println("summarizeGrade(...)");
		System.out.println(...);

	}

}
```

Output

```
Array index out of bounds.
------info()
 Student ID: 109356001
       Name: Peter
 Department: MIS
     Grades: 100 70 50 67 98
summarizeGrade(...)
Avg. Score: 77
Pass: 4, failed: 1
```

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

Reminder: Please zip the whole project

Deadline: 2020/11/16 (for Mon56) or 2020/11/17 (for Tue23)