CENG 201
Object-Oriented Programming
HOMEWORK 1

In this homework, you will create an application for assigning advisors to undergraduate students. In this application, you will create a template for the **Student** and **Advisor** objects. In order to test your application and assign advisors to students, you will create another class, named **Assignment**.

Student template should keep information about the following:

- **firstName** that contains the first name of the student.
- **lastName** that contains the last name of the student.
- **age** that contains the age of the student.
- **year** that keeps which year the student is studying. **year** can take integer values between 1 and 4.
- **GPA** of the student that can take any double value between 0 and 4, 0 and 4 included.
- **advisor** of the student keeps which advisor the student is assigned. It should keep a reference to the assigned advisor. Its value will be null unless a student is assigned to an advisor by the `assignStudentToAdvisor()`method of the Assignment class.

Advisor template should keep information about the following:

- **firstName** that contains the first name of the advisor.
- **lastName** that contains the last name of the advisor.
- A integer field **ID** that keeps the ID of the advisor. Your application should give each advisor a unique ID that starts with 1. For instance, when the first advisor is created, he should have ID 1, the second created advisor should have id 2, .. etc.
- A field named **numberOfStudentsAssigned** to keep track of how many students are assigned to the advisor
- You can keep extra attributes in your class if you need. But they should be necessary for the application, so please do not add unnecessary attributes.

Assignment class need not have any attributes. It is like a driver class with some utility functions defined in it, details of which are described in the following paragraphs.

Application Requirements

1. Default values for the age, year and GPA fields of the students are 18, 1, and 0 respectively.

2. It should be possible from the application to create a student
- with the first name and last name
- with the first name, last name and age

- with the first name, last name, age, year and GPA

3. Attributes of the student cannot be seen and modified from other classes. But these classes can access and set these fields via proper method calls.

4. It should not be possible to set year attribute a value greater than 4 or less than 1. If a wrong value like 5 is tried to be set, your application should not set that value and print the following message:

```
**************************************
ERROR: Wrong year value for student: 5
**************************************
```

"ERROR: Wrong year value for student:" part of the output is constant, and then you should print the wrong value that is tried to be set to the year field.

5. It should not be possible to set GPA attribute a negative value or a value greater than 4. If a negative value like -1 is tried to be set, your application should not set that value and print the following message:

```
**************************************
ERROR: Wrong GPA value for student: -1.0
**************************************
```

"ERROR: Wrong GPA value for student:" part of the output is constant, and then you should print the wrong value that is tried to be set to GPA.

6. There should be a method named "`toStrStudentDetails()`" in the student class that prints student information. The output of this method should be as follows:

```
**************************************
STUDENT NAME: Ayse Can
AGE: 18
YEAR: 1
GPA: 0.0
**************************************
```

7. There should be a method named "`isSuccessful()`" which will return true if the student's GPA is greater than or equal to 2.5. Otherwise it should return 0.

8. There should be a method named "`currentStatus()`" which will return the students status depending on the GPA. This method should use "`isSuccessful()`" method in order to find out whether the student is successful or not. It should print the name of the student together with its success status in the following format;

```
**************************************
STUDENT: Ahmet Can NOT SUCCESSFUL
**************************************
```

If the student's GPA is less than 2.5 than it prints NOT SUCCESSFUL after student name. Otherwise, it prints SUCCESSFUL after student's name.

9. It should be possible from the application to create an advisor

- with the first name and last name

10. firstname and lastname of the advisor cannot be seen and modified from other classes. But these classes can access and set these fields via proper method calls except .

11. ID field can only be set automatically during the creation of the Advisor object by the application.

12. **numberOfStudentsAssigned** can only be modified by `assignStudentToAdvisor()` method during the assignment of an advisor to a student.

11. There should be a method named "`toStrAdvisorDetails()`" in the advisor class that prints the advisor information. The output of this method should be as follows:

```
**************************************
ADVISOR ID: 1 NAME: Ali Can
**************************************
```

12. Assignment class performs assigning students to advisors. This is the main class of your application. There should be a method named `assignStudentToAdvisor()` in the class that assigns a student to an advisor. This method should set the advisor field of the student object that has taken as the parameter. One important requirement of your application is, **advisors can be assigned to at most 3 students.** For instance, if 3 students are assigned to an advisor and one more student is tried to be assigned, your application should not permit it, and print an error message as follows;

```
ERROR: Cannot assign more students to advisor ADVISOR ID: 1 NAME: Ali Can
```

Here in this output, similar to errors for year and GPA, "`ERROR: Cannot assign more students to advisor`" part is fixed. Then, you should print the advisor information by calling "`toStrAdvisorDetails`()" method.

13. Assignment class should have a method `whichSemester()` for calculating which semester we are in. If the current month is one of {9,10,11,12 and 1} it should return the string "FALL". If the current month is one of {2,3,4,5,6} it should return the string "SPRING". Otherwise, it should return "SUMMER".
Hint: You can use `java.util.Calendar and java.util.Date` classes. Many examples exist on the internet.

14. Both methods of advisor class should be called from the main function of Assignment class without creating instances.

15. In assignment class, write a main function and test your application for all possible function calls and object instantiation mechanisms.

16. For the test, be sure to write code for the following in your main function:

- Create 4 students, 3 advisors with different constructor calls
- Print student and advisor details
- Assign student to advisors
- Test whether more than 3 students can be assigned to an advisor

- Test whether a wrong GPA value can be assigned to a student
- Test whether a wrong year value can be assigned to a student
- Print current semester
- Print student status

IMPORTANT:

- Make sure that your files are named Student.java, Advisor.java and Assignment.java.
- Please make a good design and use static methods and attributes when you need properly.
- There is no one correct solution, but all requirements should be met. The code should be properly written such that unnecessary information and accesses to fields violating encapsulation should be prevented.
- A sample run of the application is given below, so from the main method you should create instances and test all the functionalities required from your application.
- This homework should be done individually.
- Please do it on your own, you can read tutorials and forums, many examples exist on the internet
- You should be sure to understand whatever you write in your code, I will randomly select and ask you to describe your code.
- Write good code, the one that does not smell bad.
- For your questions, use mailgroup. Ask to everybody so that everybody sees the answers. No private mails for the homework please.
- Submission details will be announced by your assistants. You will have 2 weeks to complete your work.

```
ERROR: Cannot assign more students to advisor ADVISOR ID: 1 NAME: Ali Can
**************************************
STUDENT NAME: Ayse Can
AGE: 18
YEAR: 1
GPA: 0.0
**************************************
STUDENT: Ahmet Can NOT SUCCESSFUL
**************************************
ERROR: Wrong year value for student: 5
**************************************
ERROR: Wrong GPA value for student: -1.0
**************************************
STUDENT: Ayse Can SUCCESSFUL
**************************************
Current Semester is FALL
**************************************
ADVISOR ID: 1 NAME: Ali Can
**************************************
ADVISOR ID: 2 NAME: Veli Can
**************************************
ADVISOR ID: 3 NAME: Ahmet Can
**************************************
```