KOÇ ÜNİVERSİTESİ
COMPUTER ENGINEERING

COMP 132: Advanced Programming

FALL 2022
Programming Assignment

Due: January 03, 2023, 09:00 pm (Late submissions will not be graded.)

One-to-one Tutoring Center Management System

This is an individual programming project. All work you submit must belong to you. For any questions regarding this programming project, you should use the corresponding Blackboard discussion forum and post your questions there.

You must include and sign (by writing your name and student id number) the following Pledge of Honor statement at the beginning of your main method source code file. Otherwise, your project will not be graded.

```
/************** Pledge of Honor ****************************************
I hereby certify that I have completed this programming project on my own
without any help from anyone else. The effort in the project thus belongs
completely to me. I did not search for a solution, or I did not consult any
program written by others or did not copy any program from other sources. I
read and followed the guidelines provided in the project description.

READ AND SIGN BY WRITING YOUR NAME SURNAME AND STUDENT ID

SIGNATURE: <Name Surname, Student id>
*********************************************************************/
```

Blackboard submission: You should submit one ***.zip** file named your <name-surname> (for example, Ayse-Yilmaz.zip) that contains the whole Java project folder and a report file as a PDF. The report should be written in two parts. The first part should be a guideline describing the execution steps of your application. In the second part, you should describe your project design, implemented types and their hierarchies, graphical user interfaces, and references of the resources you have used.

Please note that the report grade is part of your project grade. You should prepare it as clearly as possible. There is no page limit for the report.

Bonus: The top projects that have the best GUI design, the best report, and the best demonstration will get bonus points.

Project overview

In this project, you are expected to put your object-oriented programming (OOP) and Java Swing skills to design a graphical user interface (GUI) based one-to-one tutoring center application. The application manages information about students, courses, and tutors.

The programming project is different from the programming labs. In the labs, we provided you with all the required steps. In the project, the objective is to do large-scale programming, apply OOP concepts you have learned, do research on Swing GUI, and practice GUI programming. We provide some general constraints and leave the implementation details up to you. You are expected to write a large-scale Java application from scratch. You are expected to study the Swing API and use its GUI components in the design and implementation of your application.

This document is a guideline to describe some of the design choices while you will decide on the implementation. You should use software practices such as type hierarchies, abstraction, code cleanliness, and documentation in your work.

This is an individual project. You can use the provided course materials. You can discuss concepts with the TAs and your friends, but **cannot** share any code. The project code and report you submit **must** belong to you. You are not allowed to copy-paste code from anywhere. If you'd like to use a UI designer tool, you are expected to be aware of what all of the code does, you must describe how you used the designer in your report, explain in the source code and during the project demonstration. You must also add references to your project report. Any violation of the rules stated in the project description would not be tolerated.

In the following sections, we describe the project and specify its constraints. You should make your own decisions for anything that is not specified.

The recently established **One-to-One Tutoring Center** needs a management system for courses, students and tutors. The rules and details of the operations not mentioned are left to your design. Make sure to state your assumptions in this direction clearly. Do not try to extend the project too much by adding any other unrequired large scale features. The features that the Tutoring Center System should have are listed below.

1. Course

a. There are two types of courses: Beginner and Advanced.
    i. Beginner courses may include basic Mathematics, Introduction to Programming, Linear Algebra, Probability, and so on. (You can add more).
    ii. Advanced courses may include computer networks, parallel programming, machine learning, and so on. (You can add more).
b. A course should keep a list of its tutors. There are at least one and at most 5 tutors for a course.

Note: *Advanced courses can only be taught by A-level tutors (see the Tutor explanation).*

c. A course should keep the list of students registered for that course.
d. A course may have a pre-requisite or a list of prerequisite classes that must be taken before registering for the course.
e. A course may have one or more equipment requirements that must be bought by students before registering for the course.

Note: You may decide on the prerequisites of each course yourself. But you must keep in mind that beginner courses must <u>not</u> have an advanced course as its prerequisite).

2. Tutor

a. In addition to information related to the tutor (such as name, and ID), each tutor has a professional portrait photo and a monetary balance for earning money.
b. Each tutor keeps a time schedule for students and the courses that they teach. The assumption of scheduling is that all schedules are held in one day between 08:00-12:00 and 13:00-21:00.
c. The teaching time allocated for each student is limited to 60 minutes.
d. Each student that is being taught is registered in the list of responsible Tutors.
e. There may be different tutoring costs for each tutor, which will be used to calculate the fee of the students' tutoring.
f. Also, since each tutor has a different contract with the tutoring center, Tutors may have different percentages cut from tutoring costs gathered from students due to the system providing a platform to arrange tutoring opportunities. The percentage can be random within the boundaries specified later.

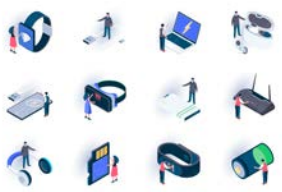
g. There are two types of Tutors:

   i. A-Level Tutors can teach both Beginner and Advanced Level courses, with **minimum** tutoring costs starting from 700 TL for each course per 60-minute session. The percentage cut from the tutoring cost can be a **maximum** of 10 percent.

   ii. B-Level Tutors can teach only Beginner courses, **maximum** tutoring cost must be 1000 TL for each course per 60-minute session. The percentage cut from the tutoring cost can be a **maximum** of 15 percent.

h. A tutor should be able to select what kind of courses he/she can teach within the qualifications of the tutor, the tutoring cost for that course based on the tutor's qualifications, select an available time for a course to teach, and inquire about a student's public information from the system at any time.

3. Equipment



a. Each piece of equipment has a price and name and a percentage that determines how much profit the tutoring center achieves by selling it. The percentage is decided beforehand by the tutoring center.

b. Some courses require pieces of equipment like a laptop, tablet, and so on.

4. Student

a. In addition to information related to the student such as name, TCKN (should be private), age, optionally a portrait photo, and so on, each student keeps a list of passed courses (if any) and courses registered currently. In order to pay for the courses and pieces of equipment, a monetary balance for each student must be kept.

b. Student should be able to change his/her information, reserve a course to take by selecting the time he/she wants to go to a specific course with which tutor, see the courses that he/she can take based on past courses he/she successfully finished and tutors' information who can teach them.

c. The student, who has been tutored, pays the expenses, including the cost of tutoring and any piece of equipment that needs to be bought for a course. The

system queries the student’s transactions and calculates the fee if a student takes courses and buys any piece of equipment.

5. Tutoring Center System

a. The System should manage the operations related to courses, tutors, and students.

b. The system is used by both tutors and students. To use the system in a meaningful manner, both students and tutors must have a username and a password to be able to login the system.

c. After performing the tutoring to a student, the corresponding tutor who taught must be able to update the student's course status for that specific course by giving a pass or fail status.

d. If the student receives a fail status for a course, which is a prerequisite for other courses, the student still is not able to take more advanced courses. But, if the student receives a pass status, then the student can take further advanced courses.

e. The system should have the capability to sell pieces of equipment to students who need to register for a course that requires those pieces of equipment with their price.

6. Administrator

a. The administrator is the manager of the whole system.

b. The administrator has both abilities of a Privileged Student (Can take every course) and an A-Grade Tutor.

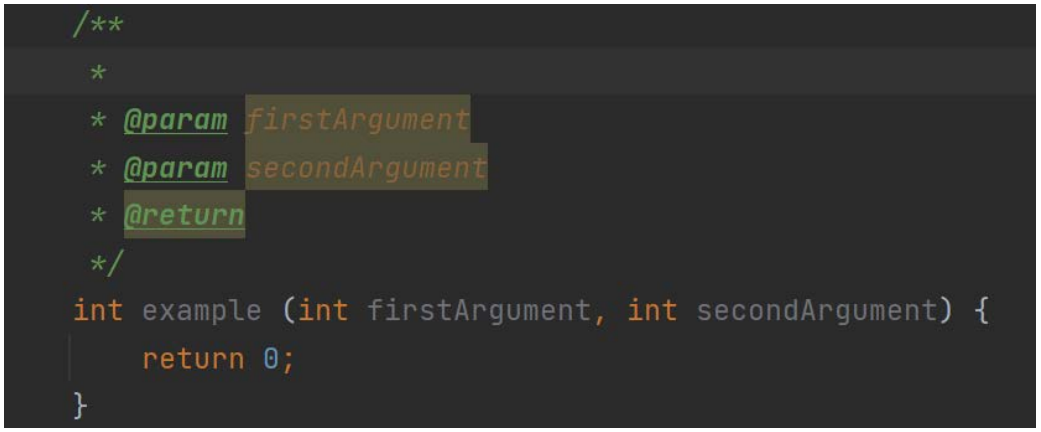
c. The administrator can add new courses to the system or change the properties of the courses available to both students and tutors.

d. The Administrator is the only one that can see the profit of the tutoring center. The profit of the tutoring center should be calculated by the system using the commission percentage of tutoring costs and equipment prices.

In your project design and implementation, you should:

- Use inheritance and type hierarchies.
- Apply polymorphism: abstract classes, interfaces.
- Use Java Collections Framework components.
- Use Java Swing GUI components.
- Apply code documentation
  - Explain what each method does, method's parameters and return type.
  - Your documentation for a method should be explanatory to a first-time reader yet concise so that a reader should understand it within a few lines. You can get inspiration from Java Language's official documentation.
  - You can also use Javadoc for documentation: https://www.oracle.com/java/technologies/javase/javadoc-tool.html

Hint: you can have your IDE generate a documentation string (docstring for short) template by typing /** above a method declaration and hitting enter.



Note: Please note that not providing sufficient documentation would lead to grade reduction, even if your code is perfect.

Design and implement a Graphical User Interface using the **Java Swing Framework**. You can use components such as **JTextField, JButton, JComboBox, and JOptionPane** to design your GUI. You are free to use any layout, but your **GUI** should be easy to understand and use.

To meet the project demo requirements, your application must *have at least*:

- 12 different courses, including both Beginner and Advanced types.
- 3 Tutors for each course (*Pay attention to the course levels and Tutors level, check the Tutor and course descriptions*).
- 8 students per course (each student can register for different courses. considering the prerequisites.)
- 5 different pieces of equipment.
- An administrator.

During the **demonstration** of your project, you should show the execution of the following operations via your application's **Graphical User Interface:**

Tutor:

- Sign-up and log in as a Tutor.
- Display/Modify Tutor information.
- Select a course and a time schedule to teach.
- Display all students' information, who took course(s) with the corresponding Tutor.

Student:

- Sign-up and log in as a Student.
- Display/modify Student information.
- Take a course and update the student's course history.

Administrator:

- Sign-up and log in as Administrator.
- Add new courses to the system.
- Display the registered students in a given course, sorted by age. In the case of the same age, they should be sorted by gender (female first).
- Display all the tutors' schedules for a given course from both students' and tutors' perspectives.
- Inquire about a student's information from a tutor, and display the student's course history.
- Calculate and charge the fee for tutoring and any required pieces of equipment to students and add the tutoring fee (with reductions due to service cost) to the tutor's balance.
- Calculate the total profit gained from each course, which specifies the equipment and tutoring profits, and display them.
- Calculate the incomes of the tutors and display them.

Good Luck!