



---

**CENG 305**

Object Oriented Programming with Java

Spring 2023-2024

**Programming Assignment 3**

---

**Due Date:** 2<sup>nd</sup> June 2024, 23:59, via ODTUCLASS

**REGULATIONS**

1. **Submission type:** You will submit a zip file named as **e1234567\_ceng305\_pa4.zip** which includes all of your BlueJ project files and generated javadoc files. e1234567 should be your student identification number. Submission will be done via **ODTUClass**. Only your last submission will be graded. If your submission fails to follow the specifications or does not compile, there will be a significant penalty in points.
2. **Late submission:** In case of late submission your score will be calculated as follows:  
**SCORE-(5\*day\*day)**
3. **Cheating: We have zero tolerance policy for cheating.** People involved in cheating will be punished according to the university regulations and will get 0 from the assignment. You can discuss algorithmic choices, but sharing code between students is strictly forbidden. Your code will be compared with those of your friends both semantically and visually. Please be aware that there are “very advanced tools” that detect if two codes are similar.
4. **No grouping:** The assignment has to be done individually.
5. **Communication:** You can use the ‘discussion forum’ on ODTUClass for your questions and share your ideas. Check the ‘news forum’ for announcements regularly. Also, you can contact with ‘atakan@ceng.metu.edu.tr’ for your problems or questions.

**SPECIFICATIONS**

**Requirements**

You are chosen by your department chair for developing **SAS (Student Automation System) Project**.

SOS will be a student management system in which the department can organize the courses by the means of adding or dropping them. An email, which the department chair has stated their expectations about SAS Project, is provided below.

“Dear Sir/Madam

Due to our departments problems with accessibility issues because of paper works, we require a student management system. The system we require, SAS project, will be comprehensive, but initially we need a system with the following features:

- The system should allow us to manage courses, instructors and students.
- Each course should have only one instructor.
- Each course should have a unique course code.
- Each student should have a unique student number.
- The system should be able to print
  - courses,
  - students for each course
  - instructors,
  - students list,
  - student info
- The system should be able to add and remove
  - student,
  - instructor,
  - course
- For each course, the system should store the following attributes :
  - Course Name
  - Course Code
  - Course Participants
  - Instructor of Course (optional)
- For each instructor, the system should store the following attributes:
  - Name Surname
  - Age
- For each student, the system should store the following attributes :
  - Name Surname
  - Age (optional)
  - Entrance year

I believe that you will create a system that will meet our requirements.

Chair of Department”

## **2.2 Design and Implementation**

After you read the email, you created a team of developers and decided to implement a system with five classes:

- Course
- Person
- Instructor
- Student
- Driver

Course is the class that represents courses. Person is the class that represents some attributes of instructors and students. Instructor is the class that represents instructors. Student is the class that represents students.

Figure 1 shows the class diagram that you prepared before starting the implementation. Sample runs are uploaded to ODTUCLASS for the final system. You should ensure that all functions in your classes match those shown in the class diagram. All functionality demonstrated in the sample runs must be implemented. Additionally, for simplicity, the "Driver" class is provided. Note that if your code is not compatible with the provided "Driver" class, there will be a significant penalty in points.

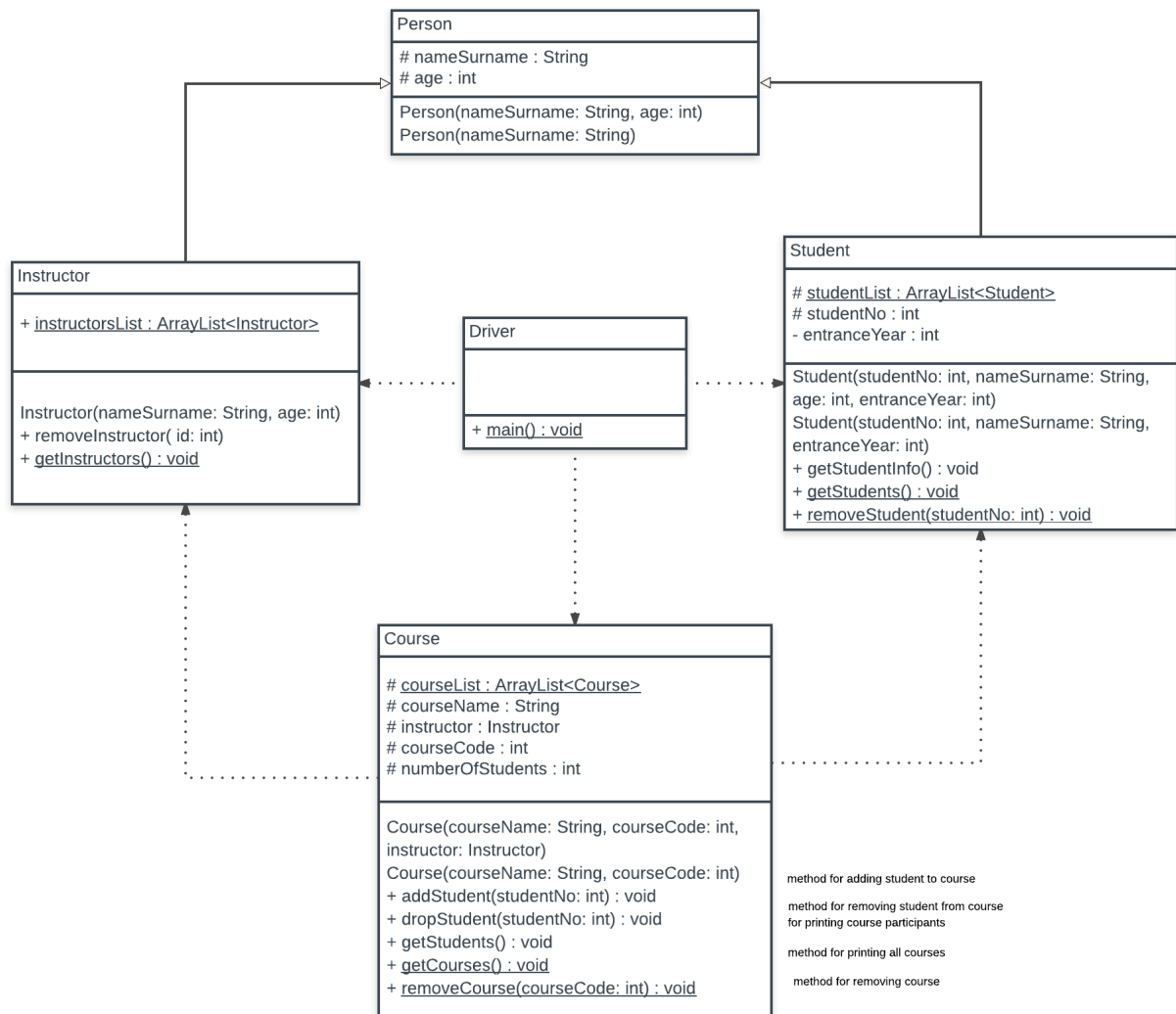


Figure 1: Class Diagram

## Project Structure

You can see the final BlueJ project structure in Figure 2.

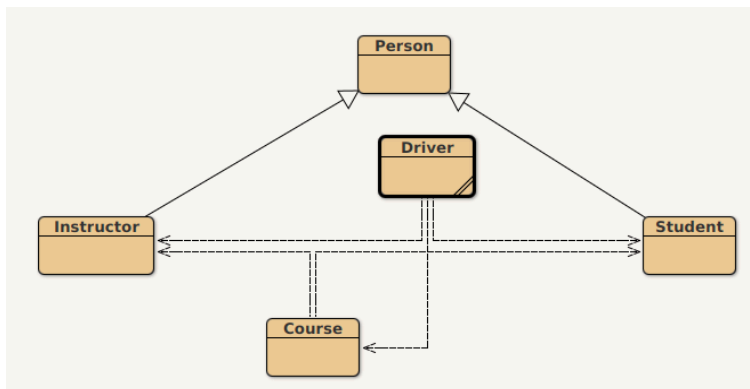


Figure 2: Project Structure