Middle East Technical University **Department of Computer Engineering**

CENG 305

Object Oriented Programming with Java

Spring 2019-2020

Homework 3

Due Date: 7th JUNE 2020, 23:59, via ODTUCLASS

REGULATIONS

1. **Programming Language:** You must use BlueJ IDE to write and implement Java code. You can download it from www.bluej.org. **You are not allowed to use any other IDE.**
2. **Submission Type:** You will submit a zip file named as e1234567_ceng305_hw3.zip which includes all of your BlueJ project files. e1234567 should be your student identification number.
3. **Late submission:** In case of late submission your score will be calculated as follows:

   SCORE-(5*days*days)
4. **Cheating:** We have zero tolerance policy with regard to cheating. People, who are involved in cheating, will be punished according to the university regulations. Your code will be compared with those of your friends both semantically and visually.
5. **No Grouping:** The assignment has to be done individually.
6. **Communication:** You can use the 'discussion forum' on ODTUCLASS for your questions and to share your ideas. Check the 'news forum' for announcements regularly. Also, you can contact with **'gozsari@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 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 department’s 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 havea 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.

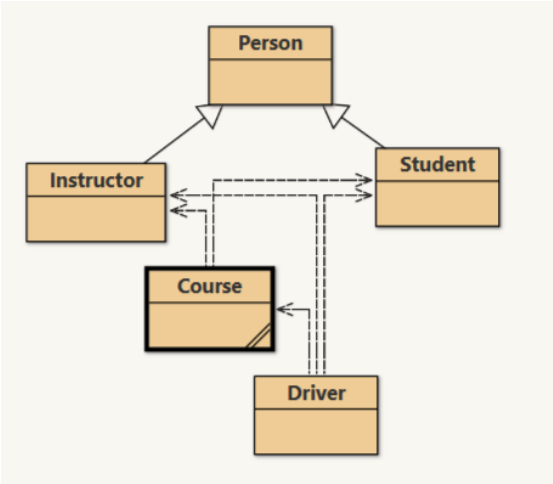
Chair of Department”

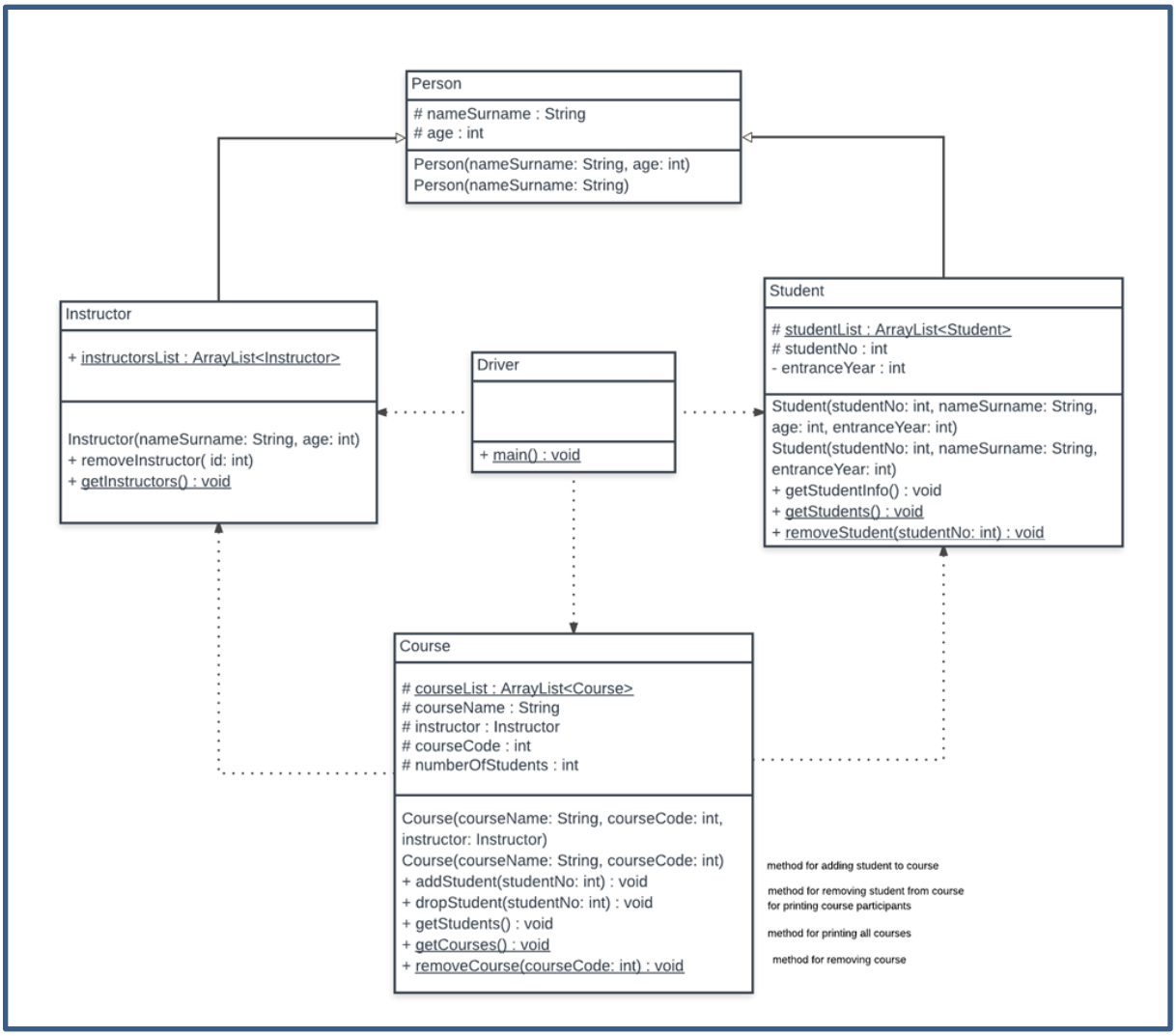
2.2 Design and Implementation

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

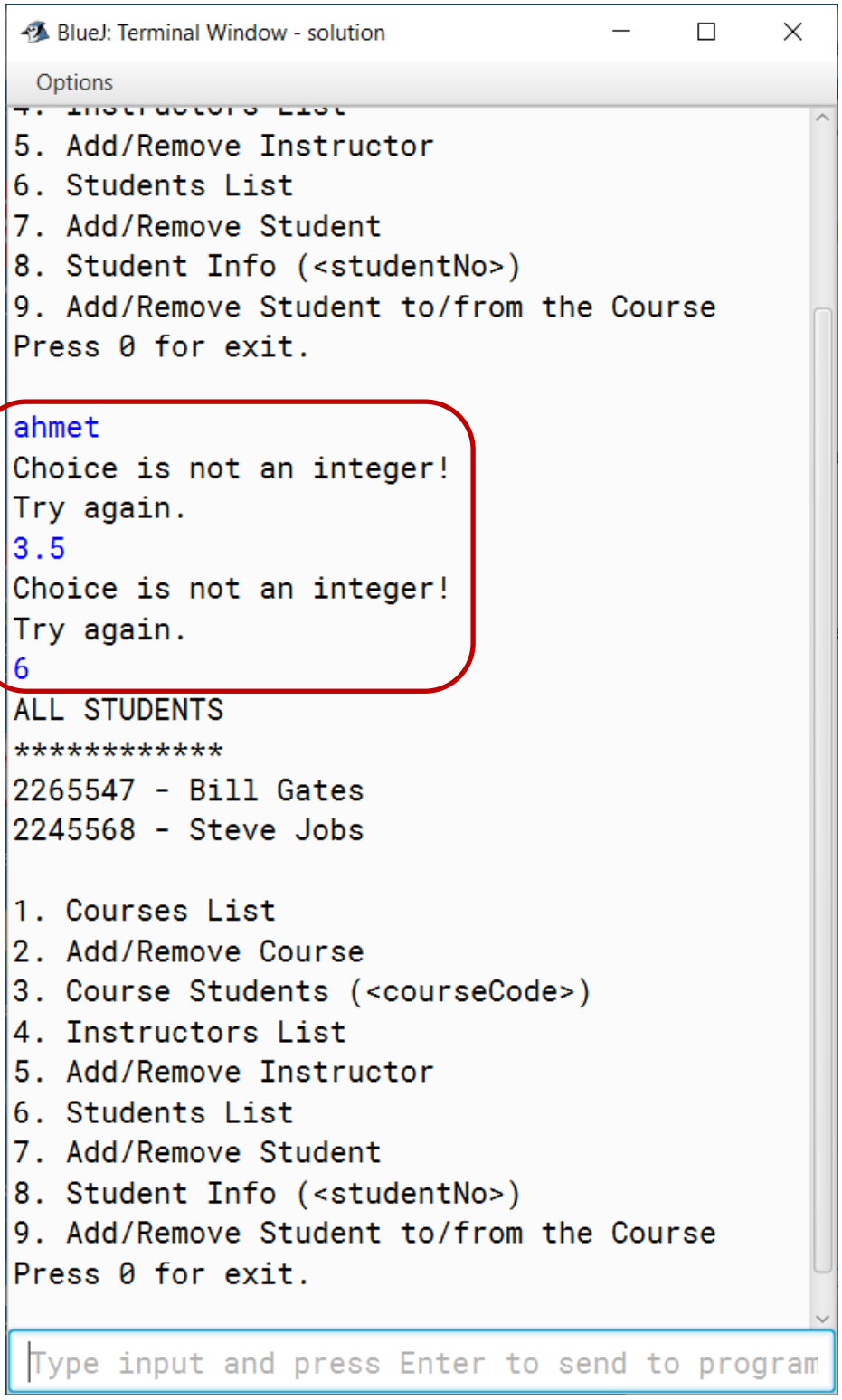
- `Course`
- `Person`
- `Instructor`
- `Student`
- `Driver`

`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.


Person
Instructor
Student
Course
Driver


Person
# nameSurname : String
# age : int
Person(nameSurname: String, age: int)
Person(nameSurname: String)
Instructor
+ instructorsList : ArrayList<Instructor>
Instructor(nameSurname: String, age: int)
+ removeInstructor(id: int)
+ getInstructors() : void
Driver
+ main() : void
Student
studentList : ArrayList<Student>
studentNo : int
- entranceYear : int
Student(studentNo: int, nameSurname: String, age: int, entranceYear: int)
Student(studentNo: int, nameSurname: String, entranceYear: int)
+ getStudentInfo() : void
+ getStudents() : void
+ removeStudent(studentNo: int) : void
Course
# courseList : ArrayList<Course>
# courseName : String
# instructor : Instructor
# courseCode : int
# numberOfStudents : int
Course(courseName: String, courseCode: int, instructor: Instructor)
Course(courseName: String, courseCode: int)
+ addStudent(studentNo: int) : void
+ dropStudent(studentNo: int) : void
+ getStudents() : void
+ getCourses() : void
+ removeCourse(courseCode: int) : void
method for adding student to course
method for removing student from course
for printing course participants
method for printing all courses
method for removing course

As it can be seen from the following figure, there must be exception handling for the choice in the menu shown below.



SAMPLE RUN:

Options

```
Student Automation Project (Enter the Choice)
*********************************************
1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

5
Add Instructor     : add # <nameSurname> # <age>
Remove Instructor  : remove # <instructor ID>
add # Dennis Ritchie # 76

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

4
ALL INSTRUCTORS
***************
ID  Name Surname
1   Dennis Ritchie

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

3
Enter course code :
5710230
INTRODUCTION TO C PROGRAMMING
*****************************************
Bill Gates
Steve Jobs
-----------------------------------*
Total 2 students at this course.
```

```
1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

9
Add Student to the Course    : add # <student number> # <courseCode>
Add Student from the Course     : remove # <student number> # <courseCode>
add # 2265547 # 5710230
Bill Gates added to INTRODUCTION TO C PROGRAMMING

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

9
Add Student to the Course    : add # <student number> # <courseCode>
Add Student from the Course     : remove # <student number> # <courseCode>
add # 2245568 # 5710230
Steve Jobs added to INTRODUCTION TO C PROGRAMMING
```

```
1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

8
Enter student no :
2265547
Student Info  : 2265547
*******************************
Name Surname  : Bill Gates
Age           : 23
Entrance Year : 2015

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

8
Enter student no :
2245568
Student Info  : 2245568
*******************************
Name Surname  : Steve Jobs
Entrance Year : 2016
```

```
1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.
7
Add Student     : add # <student number> # <name surname> # <entrance year>
Add Student     : add # <student number> # <name surname> # <age> # <entrance year>
Remove Instructor  : remove # <instructor ID>
add # 2265547 # Bill Gates # 23 # 2015

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

7
Add Student     : add # <student number> # <name surname> # <entrance year>
Add Student     : add # <student number> # <name surname> # <age> # <entrance year>
Remove Instructor  : remove # <instructor ID>
add # 2245568 # Steve Jobs # 2016

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

6
ALL STUDENTS
************
2265547 - Bill Gates
2245568 - Steve Jobs
```

```
1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

2
Add Course     : add # <courseCode> # <courseName>
Add Course     : add # <courseCode> # <courseName> # <instructor ID>
Remove Course  : remove # <courseCode>
add # 5710230 # INTRODUCTION TO C PROGRAMMING # 1

1. Courses List
2. Add/Remove Course
3. Course Students (<courseCode>)
4. Instructors List
5. Add/Remove Instructor
6. Students List
7. Add/Remove Student
8. Student Info (<studentNo>)
9. Add/Remove Student to/from the Course
Press 0 for exit.

1
ALL COURSES
***********
5710230 - INTRODUCTION TO C PROGRAMMING (Dennis Ritchie)
```