Middle East Technical University Department of Computer Engineering

CENG 443

Introduction to Object-Oriented Programming Languages and Systems

Fall 2020 - 2021

Homework 3 - Student Information System

version 1.0

Due date: 25 January 2021, 23:59

1 Introduction

In this assignment, you will implement a student information system (SIS). Once the system processes students' optical forms, it will be ready for responding to various queries about the students. When building the system, you will employ the object-oriented design principles you have learned during the classes. Furthermore, you will employ Java 8 Streams to respond to the queries efficiently.

2 Optical Forms

Optical forms are text files which are named as 1.txt, 2.txt, 3.txt, and so on. All of them reside in the **input** folder, i.e., check the image **hw3.png**. The format of an optical form is explained below.

- **1st line**: name, surname, and id number of the student are separated by a space. The student can have more than one name. And, the student id is a 7-digit number.
  Example Line: Mehmet Ali Çetin 1901834
- **2nd line**: year, course code, and course credit are separated by a space. The year is a 5-digit number. The first four digits indicate the academic year, while the last digit indicates whether it is a Fall or Sprint term (1 is for Fall term, and 2 is for Spring term). The course code is a 7-digit number. And, the course credit is a single digit number.
  Example Line: 20192 5710443 3
- **3rd line**: type of the exam. Possible choices: Midterm1, Midterm2, and Final.
  Example Line: Midterm1

- **4th line**: student's answers on the optical form interpreted by the optical form reader. The interpretation is a sequence of characters, where a character can be T (correct answer), F (wrong answer), or E (the answer not recognized by the reader).
  Example Line: FETTFTTTFTFFEETFFTTT

By looking at the sample optical form above, we can see that the student Mehmet Ali Çetin, who has the id number 1901834, took a course with code 5710443 and worth 3 credits in the Spring term of the academic year 2019-2020. And, he took 50 points from the first midterm (since there are 20 characters in the sequence, each question worths 5 points).

3 Queries

When testing your program, we will first instantiate your student information system: **SIS informationSystem = new SIS();** Then, we will call its methods that you are to implement.

- **getGrade(int studentID, int courseCode, int year)**: Returns student's overall grade for the course offered in the year. The overall grade is 0.25 x Midterm1 + 0.25 x Midterm2 + 0.5 x Final.
  Example Returned Double Value:
  72.5

- **updateExam(int studentID, int courseCode, String examType, double newGrade)**: Updates student's exam. Other method calls should be affected after calling this method (i.e., the old grade is not considered anymore). The student can take the same course in different years. If that is the case, the most recent one's exam grade will be updated.

- **createTranscript(int studentID)**: Prints the transcript of the student. Example Output (separated by a space; sorted by a year then a course code):

  ```
  20101
  2360119 DD
  5710111 AA
  20102
  2360119 BB
  2360120 CB
  ...
  20132
  5710492 BA
  CGPA: 3.52
  ```

  For score to letter grade conversion (i.e., 92 → AA), use our university's course credit system. CGPA calculation is basically weighted average of letter grades where the weights are the course credits. Finally, if the student took the same course before, only the most recent one contributes to the CGPA calculation.

- **findCourse(int courseCode)**: Prints the years when the course was offered as well as the number of registered students.
  Example Output (separated by a space; sorted by a year):

  ```
  20181 44
  20182 40
  20191 51
  ```

- **createHistogram(int courseCode, int year)**: Prints the grade histogram of the registered students in the course offered in the year.
  Example Output (separated by a space; sorted by a bin value):

  ```
  0-10 0
  10-20 5
  20-30 2
  30-40 3
  40-50 1
  50-60 10
  60-70 3
  70-80 6
  80-90 1
  90-100 2
  ```

  The former value in the bin value is inclusive, while the latter is exclusive. As an example, 10-20 means [10, 20).

4 Specifications

- createTranscript(), findCourse(), and createHistogram() prints their results to **stdout**.
- You will only complete parts where **TODO** comments are.
- You are only allowed to declare local variables in the methods, i.e., do not declare new attributes/methods in class SIS (Since the query types are known in advance, the format of the optical form can be exploited, and implementing the methods becomes trivial).
- Do not use loops instead of Java 8 Streams. Your code will be inspected and be penalized heavily if you do so (-50 pts).

5 Grading Rubric

- Commented Code: 10 pts
- getGrade(): 15 pts

- updateExam(): 15 pts
- createTranscript(): 30 pts
- findCourse(): 15 pts
- createHistogram(): 15 pts

6 Submission

Submission will be done via ODTUClass. You will submit a zip file called **hw3.zip** that contains only **SIS.java**. A penalty of **5 x LateDay x LateDay** will be applied for submissions that are late at most 3 days.