

PROGRAMMING ASSIGNMENT 3

TAs : Alpay TEKİN

Due Date : 18.05.2025 (23:00:00)

Programming Language: Java 8 (Oracle)

University Student Management System



1 Introduction

You are tasked with developing a Student Management System that can handle the storage, processing, and reporting of academic data, including students, faculty members, courses, departments, and academic programs. The system must read data from various input files, create relevant entities, establish meaningful relationships among them, and generate detailed reports in a structured and human-readable format.

Your implementation must follow the fundamental principles of object-oriented programming (OOP). Specifically, your design should demonstrate the use of inheritance, encapsulation, polymorphism and abstraction.

2 System Definition

In this part, you will find the basic attributes of the users and items. You must implement at least those attributes.

2.1 Person

There are 2 types of person: academic member and student. You must consider followings:

- Each person type has **unique** ID, name, email and department (CS, AI etc.).
- Each student can enroll multiple courses.

- Each academic member can teach multiple courses.
- Only one academic member can assigned as head of the department.

2.2 Academic Entity

There are 3 types of academic entity: course,program and department. Programs can contain multiple courses.

- Each academic entity type has code,name and description.
- Each department is headed by a faculty member.
- Each course has credit and semester. Courses are gradable, valid letter notes are A1,A2,B1,B2,C1,C2,D1,D2 and F3. You can find the corresponding values of letter grades in the 4-point grading system on in table 1.

Letter Grade	4-Point System Value
A1	4.00
A2	3.50
B1	3.00
B2	2.50
C1	2.00
C2	1.50
D1	1.00
D2	0.50
F3	0.00

Table 1: Conversion of Letter Grades to 4-Point Grading System

2.3 Student Management System

The StudentManagementSystem class serves as the core controller for managing academic entities within the application. It is responsible for storing, processing, and organizing information related to students, faculty members, departments, programs, and courses. System provides method to:

- Establish relationships among entities (e.g., course enrollments, faculty assignments)
- Assign and process grades
- Generate structured reports for students and courses.

3 Specifications

The system must provide the following functionalities:

- Assign a faculty member as the head of a department.
- Associate courses with their corresponding academic programs.
- Enroll students in designated courses.

- Calculate the average grade and grade distribution for each course and present them in a well-structured report. The report should also include general course information (code, name, department, credits, semester, instructor) along with the list of enrolled students. Grade distribution presents the count of occurrence of letter grades. To compute the average grade for a course, sum all the grades using the 4-point grading system and divide the total by the number of students.
- Calculate the GPA for each student and present it in a well-structured report. The report should also include general student information (ID, name, email, major, status) along with the lists of enrolled and completed courses. To compute a student's GPA, multiply each course grade (in the 4-point grading system) by its corresponding credit, sum up these values, and then divide the total by the sum of all credits.
- System must handle following exceptions:
 - Invalid person type
 - Non-existent student
 - Non-existent academic member
 - Non-existent department
 - Non-existent program
 - Non-existent course
 - Invalid letter grades

4 Definition of Input and Output

The program will read input data from text files, including person information, departments, programs, courses, assignments (indicates assignment of course to instructor and students) and grades. **The some of the example input and output files shared in this section are not provided in their complete form. The full versions will be shared via Piazza.**

4.1 Inputs

The person.txt file contains type of the person, name, email and major/department. Each entry follows a structured format:

- For Students, <Item Class>,<ID>,<Name>,<Email>,<Department>.
- For Academic members, <Item Class>,<ID>,<Name>,<email>,<Major>.

The department.txt file complies with following format:

- <code>,<name>,<description><head>

The program.txt file complies with following format:

- <code>,<name>,<description><department><degreeLevel><totalCredits>

The courses.txt file complies with following format:

S,101,Benno Marco,marco@uni.edu,CS
S,102,Hiroko Reiko,reiko@uni.edu,CS
S,103,Dimitriy Kazimir,kazimir@uni.edu,CS
S,104,Apollon Charis,charis@uni.edu,CS
S,105,Mehmet Ozan,ozan@uni.edu,CS
S,201,Nuno Isaias,isaias@uni.edu,AI
S,202,Hildegard Stellan,stellan@uni.edu,AI
S,203,Roksolana Klavdiya,klavdiya@uni.edu,AI
S,204,Filimon Pompiliu,pompiliu@uni.edu,AI
S,205,Malkhazi Pridoni,pridoni@uni.edu,AI
F,101,Prof.Dr. Iyas Cicekli,cicekli@uni.edu.tr,AI
F,102,Prof.Dr. Pinar Sahin,sahin@uni.edu,AI
F,103,Prof.Dr. Erkut Erdem,erkut@uni.edu,AI
F,104,Prof.Dr. Nazli Kizler,ikizler@uni.edu,AI
F,201,Prof.Dr. Ahmet Burak Can,burakcan@uni.edu.tr,CS
F,202,Prof.Dr. Lale Ozkahya,ozkahya@uni.edu,CS
F,203,Prof.Dr. Suat Ozdemir,ozdemir@uni.edu,CS

Table 2: Example Input Format for Person

CENG,Computer Engineering,The Department of Computer Engineering,101
--

Table 3: Example Input Format for departments

BBM,Computer Engineering,The Computer Engineering (BBM) program,Computer Engineering,Bachelor,50
AIN,Artificial Intelligence Engineering,The Artificial Intelligence Engineering (AIN) program,Computer Engineering,Bachelor,40

Table 4: Example Input Format for Programs

- `<code>,<name>,<department><credits>,<semester>,<programCode>`

The assignments.txt file complies with following format:

- `<personType>,<ID>,<courseCode>`

The grades.txt file complies with following format:

- `<letterGrade>,<studentID>,<courseCode>`

4.2 Output

The program will write the output to a file, and the filename will be provided as a command-line argument. Users must strictly follow the specified file format, as the correctness of the output will be verified using the diff command. More sample inputs and outputs will be shared on Piazza.

```

AIN440,Introduction to Natural Language Processing,Computer Engineering
,15,2024/Spring,AIN
AIN312,Formal Languages and Automata Theory,Computer Engineering,12,2024/
Spring
AIN434,Fundamentals of Computational Photography,Computer Engineering
,15,2024/Spring
AIN211,Principles of Artificial Intelligence,Computer Engineering
,18,2024/Spring,AIN
AIN431,Introduction to Computer Vision,Computer Engineering,12,2024/
Spring
BBM402,Theory of Computation,Computer Engineering,12,2024/Spring,AIN
BBM405,Fundamentals of Artificial Intelligence,Computer Engineering
,15,2024/Spring,BBM
BBM458,Wireless and Mobil Networks,Computer Engineering,12,2024/Spring,
BBM
BBM444,Fundamentals of Computational Photography,Computer Engineering
,15,2024/Spring,BBM
BBM406,Fundamentals of Machine Learning,Computer Engineering,12,2024/
Spring,BBM
BBM495,Introduction to Natural Language Processing,Computer Engineering
,18,2024/Spring,BBM

```

Table 5: Example Input Format for Courses

```

F,101,AIN440
F,101,AIN312
F,103,AIN434
F,102,AIN211
F,104,AIN431
F,202,BBM402
F,102,BBM405
F,203,BBM458
F,103,BBM444
F,201,BBM406
S,101,AIN440
S,102,AIN440
S,103,AIN440
// continue with other assignments

```

Table 6: Example Input Format for Courses

Example Output

```

Reading Person Information
Reading Departments
Reading Programs
Reading Courses
Reading Course Assignments
Reading Grades

```

```
A1,201,AIN440
A1,202,AIN440
A2,203,AIN440
A2,204,AIN440
A2,205,AIN440
A2,101,AIN440
B1,102,AIN440
B1,103,AIN440
B2,104,AIN440
B2,105,AIN440
// continue with other grade assignments
```

Table 7: Example Input Format for Courses

```
-----
                        Academic Members
-----

Faculty ID: 101
Name: Prof.Dr. lyas      iekli
Email: cicekli@uni.edu.tr
Department: AI

Faculty ID: 102
Name: Prof.Dr.P nar     ahin
Email: sahin@uni.edu
Department: AI

% continue with other faculty members

-----

                        STUDENTS
-----

Student ID: 101
Name: Benno Marco
Email: marco@uni.edu
Major: CS

Student ID: 102
Name: Hiroko Reiko
Email: reiko@uni.edu
Major: CS

% continue with other students

-----
-----
```

DEPARTMENTS

Department Code: CENG
Name: Computer Engineering
Head: Prof.Dr. lyas iekli

PROGRAMS

Program Code: AIN
Name: Artificial Intelligence Engineering
Department: Computer Engineering
Degree Level: Bachelor
Required Credits: 40
Courses: {AIN440,AIN211,BBM402}

Program Code: BBM
Name: Computer Engineering
Department: Computer Engineering
Degree Level: Bachelor
Required Credits: 50
Courses: {BBM405,BBM458,BBM444,BBM406,BBM495}

COURSES

Course Code: AIN211
Name: Principles of Artificial Intelligence
Department: Computer Engineering
Credits: 18
Semester: 2024/Spring

Course Code: AIN312
Name: Formal Languages and Automata Theory
Department: Computer Engineering
Credits: 12
Semester: 2024/Spring

% continue with other courses

COURSE REPORTS

Course Code: AIN211
Name: Principles of Artificial Intelligence

Department: Computer Engineering
Credits: 18
Semester: 2024/Spring

Instructor: Prof.Dr.P nar ahin

Enrolled Students:

- Nuno Isaias (ID: 201)
- Hildegard Stellan (ID: 202)
- Roksolana Klavdiya (ID: 203)
- Filimon Pompiliu (ID: 204)
- Malkhazi Pridoni (ID: 205)

Grade Distribution:

Average Grade: 0.00

Course Code: AIN431
Name: Introduction to Computer Vision
Department: Computer Engineering
Credits: 12
Semester: 2024/Spring

Instructor:

Enrolled Students:

- Nuno Isaias (ID: 201)
- Hildegard Stellan (ID: 202)
- Roksolana Klavdiya (ID: 203)
- Filimon Pompiliu (ID: 204)
- Malkhazi Pridoni (ID: 205)

Grade Distribution:

A1: 2

B1: 2

B2: 1

Average Grade: 3.30

Course Code: AIN440
Name: Introduction to Natural Language Processing
Department: Computer Engineering
Credits: 15
Semester: 2024/Spring

Instructor: Prof.Dr. lyas iekli

Enrolled Students:

- Benno Marco (ID: 101)
- Hiroko Reiko (ID: 102)
- Dimitriy Kazimir (ID: 103)
- Apollon Charis (ID: 104)
- Nuno Isaias (ID: 201)
- Hildegard Stellan (ID: 202)
- Roksolana Klavdiya (ID: 203)
- Filimon Pompiliu (ID: 204)
- Malkhazi Pridoni (ID: 205)

Grade Distribution:

A1: 2
A2: 4
B1: 2
B2: 1

Average Grade: 3.39

Course Code: BBM402
Name: Theory of Computation
Department: Computer Engineering
Credits: 12
Semester: 2024/Spring

Instructor: Prof.Dr.Lale zkahya

Enrolled Students:

- Benno Marco (ID: 101)
- Hiroko Reiko (ID: 102)
- Dimitriy Kazimir (ID: 103)
- Apollon Charis (ID: 104)
- Mehmet Ozan (ID: 105)

Grade Distribution:

A1: 1
A2: 1
C1: 3

Average Grade: 2.70

STUDENT REPORTS

Student ID: 101
Name: Benno Marco
Email: marco@uni.edu

Major: CS

Enrolled Courses:

- Fundamentals of Machine Learning (BBM406)
- Introduction to Natural Language Processing (BBM495)

Completed Courses:

- Introduction to Natural Language Processing (AIN440): A2
- Theory of Computation (BBM402): A2

GPA: 3.50

Student ID: 102

Name: Hiroko Reiko

Email: reiko@uni.edu

Major: CS

Enrolled Courses:

- Fundamentals of Machine Learning (BBM406)
- Introduction to Natural Language Processing (BBM495)

Completed Courses:

- Introduction to Natural Language Processing (AIN440): B1
- Theory of Computation (BBM402): A1

GPA: 3.44

4.3 Error Cases

You can find an example output that demonstrates how to handle the possible errors described in the previous sections using Java exceptions.

Example Output

```
Reading Person Information
Invalid Person Type
Reading Departments
Academic Member Not Found with ID 101
Reading Programs
Reading Courses
Program BM Not Found
Reading Course Assignments
Academic Member Not Found with ID 101
Academic Member Not Found with ID 103
Course BBM495 Not Found
Student Not Found with ID 105
```

```
Student Not Found with ID 201
Reading Grades
Course BBM495 Not Found
Student Not Found with ID 106
The grade A3 is not valid

//continue with other information presented in output section
```

5 Test and Execution

Your code must be compiled and executed under Java 8 and dev.cs.hacettepe.edu.tr. If your code does not compile and execute under developer server, then you will be graded as 0 for code part even if it works on your own machine. Your program must take input files, and name of the output file. Sample run command is as follows:

```
javac Main.java
java Main persons.txt departments.txt programs.txt courses.txt
      assignments.txt grades.txt output.txt
```

6 Grading Policy

Task	Grade
Class Implementations using OOP Principles	40
Interface Implementation	20
Correct Output	30*
Comments in JavaDoc Style	10**
Total	100

***Even though producing correct output and commenting seems like enough to get full credit, you must obey to the given rules in the PDF (for example using concept of OOP and four pillars of OOP etc.) otherwise you may face with some point deductions which may result with a grade that is as low as zero. There will be two overall multipliers about quality of your OOP and clean code separately which will vary in between 0 and 1! Note that there may be any other multipliers or point deductions in case of violation of the rules.**

**** The score of the clean code comment part will be multiplied by your overall score (excluding clean code comment part) and divided by the maximum score that can be taken from these parts. Say that you got 45 from all parts excluding clean code comment part and 10 from clean code comment part, your score for clean code comment part is going to be $10 \cdot (45/90)$ which is 5 and your overall score will be $45 + 5 = 50$.**

7 Submit Format

File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system).

```
-b<studentID>.zip  
-Main.java, or *.java
```

Late Policy

You have two days for late submission. You will lose 10 points from maximum evaluation score for each day (your submitted study will be evaluated over 90 and 80 for each late submission day). You must submit your solution in at the most two days later than submission date, otherwise it will not be evaluated. Please do not e-mail to me even if you miss the deadline for a few seconds due to your own fault as it would be unfair for your friends, e-mail submissions will not be considered if you do not have a valid issue.

Notes and Restrictions

- Your code must be able to execute on our department's developer server (dev.cs.hacettepe.edu.tr).
- You must use JavaDoc commenting style for this project, and you must give brief information about the challenging parts of your code, do not over comment as it is against clean code approach. Design your comments so that if someone wants to read your code they should be able to easily understand what is going on. You can check here to access Oracle's own guide about JavaDoc Sytle.
- **Use inheritance,encapsulation,polymorphism and abstraction where it makes logical and practical sense. Otherwise, you risk losing points for not properly applying OOP principles.**
- You must obey given submit hierarchy and get score (1 point) from the submit system.
- Do not miss the submission deadline.
- You can benefit from Internet sources for inspiration but do not use any code that does not belong to you.
- You can discuss high-level (design) problems with your friends but do not share any code or implementation with anybody.
- Source code readability is a great of importance. Thus, write READABLE SOURCE CODE, comments, and clear MAIN function. This expectation will be graded as "clean code".
- Use UNDERSTANDABLE names for your variables, classes, and functions regardless of the length. The names of classes, attributes and methods must obey to the Java naming convention. This expectation will be graded as "coding standards".
- You can ask your questions through course's Piazza group, and you are supposed to be aware of everything discussed in the Piazza group. General discussion of the problem is

allowed, but **DO NOT SHARE** answers, algorithms, source codes and reports.

- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.