

Homework 1

CENG431 – Building Software Systems

In this homework, you are expected to implement an application called “**Multilingualism**” in Java. You should fulfill the concepts:

- Object Orientation Fundamentals,
- Abstract Data Types
- Inheritance, Polymorphism, Abstract Classes, Interfaces, Exceptions
- Collections Generics
- CSV file I/O

In Multilingualism, the users of the application choose a language to learn and start taking quizzes about the language. The users are assumed to speak English and are expected to choose one of the languages between Turkish, German, Italian, or Spanish.

In this application, there are a random number of units (between 60-100) of each language such as Unit 1, Unit 2, etc. Each unit has a random number of quizzes (between 1-10). Additionally, there are a random number of questions (between 8 and 15) in each quiz. Each question can be one of four types of questions which are reading, listening, speaking, and word-matching. Also, each question type has a different point: reading is 10 points, listening is 7 points, speaking is 8 points, and word-matching is 5 points.

When a user takes a quiz, whether the user answers the questions correctly or not (for each question separately) are randomly determined. At the end of the quiz, the user wins points for each question he/she answers correctly. For example, a user takes a quiz of 8 questions. There are 3 reading questions, 1 word-matching question, 2 listening questions, and 2 speaking questions. The user answers 2 reading questions, 1 word-matching question, and 1 speaking question correctly. The user gets 33 points in total for this quiz.

There may be String and/or Audio in different question types. The String objects will be created randomly and do not have to have a meaning or be in any language at all. The Audio has a randomly created length in seconds. Here are the rules for the different types of questions:

1. A reading question consists of two strings: one is assumed to be in English, and the other is assumed to be its translated version of the chosen language.
2. A listening question consists of one string that is assumed to be in the chosen language and audio.
3. A speaking question consists of two audios.
4. A word-matching question consists of a matching of strings where the key strings are assumed to be in English, and the value strings are assumed to be their translated versions of the chosen language.

In this application, there are Bronze, Silver, Gold, Sapphire, and Ruby leagues for each language. When a user first starts to use the application, he/she is in the Bronze league. For a user to advance to the Silver League, he/she must be in the top 15 of the points ranking. To be advanced to the Golden League, a user is expected to be in the top 10 of the points rankings.

For a user to be advanced to the Sapphire League, it is expected that she/she must have at least a streak of 7 days and is in the top 5 of the point rankings. To be advanced to the Ruby league, a user must have at least a streak of 30 days and collect more than 5000 points or be in at least Unit 10 of the selected language. Each user's number of days in his/her streak is determined randomly (between 0 and 365).

When the application is first run, the number of units of each language, the number of quizzes in each unit, and the number and the types of questions in each quiz are determined randomly and written in a CSV file called "languages.csv". This file SHOULD be created once, and it SHOULD NOT change each time the application is run.

In this homework, the user data file should be read for usernames and passwords; the language choice and the number of days in the streak of each user are determined randomly to create user objects. Also, the number of quizzes that the user will take (during the current run) is determined randomly (between 6- the total number of quizzes in the chosen language) and is used to determine the user's current Unit number.

In this homework, you are expected to implement the necessary classes to load the data from the given CSV file and create the desired queries.

After creating the user, quiz, and other necessary objects, please display the results of the following queries:

- 1- The user who has the maximum points.
- 2- The user who is in the most advanced Unit in German (Hint: Unit 10 is more advanced than Unit 9)
- 3- The language which has the maximum number of units
- 4- The language which has the maximum number of quizzes
- 5- The top three users in the Silver league for Italian

If the result of any of the queries is more than one, please display all results. Since almost all data are randomly determined, the results of your projects will differ from each other, and this is normal.

Please record the chosen language, the current Unit, the number of solved quizzes, and the total points to "users.csv" after each run (they may be changed after each run).

Important Notes:

1. **Do NOT request inputs in your app.** Printing the results of the queries will be enough. An example of an **imaginary** output is given below:

```
1- bkrfrazvih 6369 points
2- rvaaqilojl Unit 88
3- German 99 Units
4- Turkish 987 Quizzes
5- Italian Silver League Top 3: 1.bxwxcitvua 2.sflhrwhwrc 3.nedzhseybm
```

2. You should use relative paths (e.g. Files/sample.csv) instead of absolute paths (e.g. C:\\user\\eclipse-workspace\\MyProject\\Files\\sample.csv).

3. To support **Turkish characters**, you may need to change your project's text file encoding to UTF8: Right-click on your project (in package explorer) → Properties → Text file encoding → Other → UTF8 → Apply.

4. You are expected to write clean, readable, and tester-friendly code. Please try to maximize reusability and prevent redundancy in your methods.

Assignment Rules:

1. In this lecture's homework, there are no cheating allowed. If any cheating has been detected, they will be graded as 0 and there will be no further discussion on this.
2. You are expected to submit your homework in groups. Therefore, only one of you will be sufficient to submit your homework.
3. Make sure you export your homework as an **Eclipse project**. You can use other IDEs as well; however, you must test if it is supported by Eclipse. If the project import is not accomplished in Eclipse, you will lose points.
4. Submit your homework through Cloud-LMS.
5. Your exported Java Project should have the following naming format with your assigned group ID (which will be announced on MS Teams) as given below:

G05_CENG431_HW1

Also, the zip folder that your project is in should have the same name

G05_CENG431_HW1.zip

6. Please beware that if you do not follow the assignment rules for exporting and naming conventions, you will lose points.
7. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.