

# CENG211 – Programming Fundamentals

## Homework #3

In this homework you are expected to implement a “**Content Rating Company**” and simulate it in Java. You should fulfill the concepts of:

- Interfaces
- Abstract Classes
- Abstract Data Types
- CSV File I/O

In the simulation, you are expected to build a company that rates the given games and movies and produce a rating output.

The content that will be rated can be either a movie or a game. There are three types of games which are Indefinite Games, Story Games, Casual Games. CSV format for the contents will be as follows:

- Movie: arrival day, 0, name, year, duration(min), average rating
- Indefinite Game: arrival day, 1, name, minimum evaluation time(hour), average rating
- Story Game: arrival day, 2, name, story duration(hour), average rating
- Casual Game: arrival day, 3, name, match duration(hour), average rating

The workers in the company can be either movie critic or game critic. The rating calculation of critics will be as follows:

- Movie:  $\text{average rating} + ((\text{duration} - 150) * 0.01) - ((2021 - \text{year}) * 0.01) + (\text{critic's opinion})$
- Indefinite Game:  $\text{average rating} + ((10 - \text{minimum evaluation time}) * 0.25) + (\text{critic's opinion})$
- Story Game:  $\text{average rating} + (\text{story duration} * 0.25) + (\text{critic's opinion})$
- Casual Game:  $\text{average rating} + ((\text{match duration} - 3) * 3) + (\text{critic's opinion})$

Simulation:

As a company, you are receiving games and movies to rate them for each day. Also, you have 3 movie critics and 5 game critics (8 in total).

The simulation will work for 5 days. Critics will be assigned to a game or a movie. They should complete their evaluation in given time before moving on to the next one. Last content that arrives to the company should be evaluated first so contents should be evaluated in LIFO (last in first out) order (Movie and Games are separate here) (First day will be given as 1 at arrival day value in the csv file). Critics will wait for assignment in a given order. If a critic completes its assignment, he/she will be wait at the last of the queue so critics should be assigned to a content in FIFO (first in first out) order.

Rating for a movie is between 0.0 and 10.0 (e.g., 7.6); rating for a game is between 0 and 100 (e.g., 87). Ratings should be sorted by name (movies and games will be sorted separately).

Note that: All of the movies or games may not be evaluated in 5 days.

Rules:

- A movie critic can watch only 1 movie in a day.
- A game critic can work at most 8 hours a day.
- A game critic should play indefinite games for 4 hours.
- A game critic should play casual games for 3 matches.
- A game critic should play story games for complete story duration.

Order of the critics and their opinions:

1. Movie critic = +0.1
2. Movie critic = -0.2
3. Movie critic = +0.3

1. Game critic = +5
2. Game critic = +9
3. Game critic = -3
4. Game critic = +2
5. Game critic = -7

Expected output format:

1. day:

1. movie critic evaluated (#movie name)
2. movie critic evaluated (#movie name)
1. game critic works on (#game name)
2. game critic works on (#game name)
3. game critic works on (#game name)
4. game critic works on (#game name)
3. game critic evaluated (#game name)
1. game critic evaluated (#game name)
- .
- .
- .

2. day:

3. movie critic evaluated (#movie name)
1. movie critic evaluated (#movie name)
2. movie critic evaluated (#movie name)
5. game critic works on (#game name)
1. game critic works on (#game name)
- .
- .
- .

3. day:

....

4. day:

...

5. day:

...

Ratings:

#Movie name (#year), rating

#Movie name (#year), rating

.

.

.

#Game name, rating

#Game name, rating

.

.

.

### Important Notes:

1. You can use standard **java.io** packages to read files. Do NOT use other 3<sup>rd</sup> party libraries.
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 from 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 supported by Eclipse.
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 the given below:

**G05\_CENG211\_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.