CSCI 3381 OO with Java   Project 1

Object-Oriented Design and Implementation of a Netflix Storage & Suggestion Data Back-End

Background:

A data back-end stores and gives access to data. For this project you store weekly top 10 Netflix data (data file provided for you). You will also provide functional access to the data. Mainly you are programming what Netflix already does. When someone finishes a show, Netflix suggests another to watch.

Data downloaded from https://top10.netflix.com/ (look at the bottom and find "Download Global List .tsv")
Original format:

week	category	weekly rank	show titles	season title	weekly hours viewed	cumulative weeks in top 10
2022-09-04	Films (English)	1	Me Time	N/A	56560000	2
2022-09-04	Films (English)	2	Love in the Villa	N/A	41220000	1
Etc…						

Text File format (netflixTopTenProcessed.txt):

```
2022-09-04
Films (English)
1
Me Time
N/A
56560000
2
2022-09-04
Films (English)
2
Love in the Villa
N/A
41220000
1
Etc…
```

Object-Oriented design:
The single entity class should be a show in a given week. I might call this ShowInWeek. This includes the statistics seen above. So, the film "Me Time" in the week of 2022-09-04 would be one instance. Collection class has all shows in their given week. If it was just the text file above, there would be two items in the Collection class.

Functionality of the Collection class:

- Add new data: https://top10.netflix.com/ updates weekly. Your program should allow a new single entity instance to be manually created and added to the collection. This is just hard coded, not entered from a file or from the keyboard.
- Purge a show: This would happen if Netflix lost the right to show something. This should not remove any data from the collection, rather mark every ShowInWeek with that name to prevent it from being suggested.
- Unpurge a show: This means Netflix has regained the right to show something and purging should be reversed.
- Edit a show. All data in a given ShowInWeek can be modified. (Setters).
- Random Suggestion: Randomly suggest something to watch.
- Predictive Suggestion: given a ShowInWeek suggest a show to watch. This can be a meaningful prediction or not depending on your intellectual curiosity. There is no test of accuracy, but you should not suggest the given show.

- Predictive Suggestion: given a collection of ShowInWeek(s) suggest a show to watch. This can be a meaningful prediction or not depending on your intellectual curiosity. There is no test of accuracy, but you should not suggest any of the given shows.
- Get Shows: Given a date (week), return a subset collection of all shows in that week.
- Any other functionality you would like to add!!!!

Design

To build the Netflix Storage & Suggestion Data Back-End system, you will need to create several classes that work together with appropriate object-oriented design. You might use the Complex Roster project as inspiration, but your project should not use inheritance. You will need to have the following parts:

- Text file to store data between runs. It should only be read at the very start of a program run and written to at the very end of a program run.
- The two classes outlined above. These should follow good Object-Oriented design.
- A tester class that systematically tests the system.

THIS IS NOT MEANT TO BE A WORKING APPLICATION. Meaning don't make this with a user in mind. Your tester class will not have any user input from the keyboard. It is only to systematically test that all functionality of the single entity and collection classes work. A portion of the tester might be the following:

```
public static void main(String[] args) {
  Shows allData = new Shows("./project1/netflixTopTenProcessed.txt");
  ShowInWeek newSIW = new ShowInWeek("2022-09-11","cat",1,"show title","season title",2500,1);
  allData.add (newSIW);
  Shows oneWeek = allData.getWeek ("2022-09-04");
  System.out.println(oneWeek);

  // etc. etc.
  allData.writeToFile();
}
```

Project Objective: in completing this project, you will

- Enhance your understanding of Object-Oriented design.
- Build a Back-End data management set of classes that meets the project requirements.
- Create and Implement a Testing Plan (tests all functionality of the individual and collection classes.

Project #1

Due September 29th via GitHub and Blackboard. Submit your project by adding me as a contributor to the project in your GitHub account.

Honor:

This is an independent programming project, and it is very important that you understand and abide by the policy concerning programming projects. Remember, your personal honor and integrity is far more important than your grade on the project.