NT Lab 12: Classes & Objects (100 points)

Objectives

- Practice defining an object class, and creating and using objects
- Practice reading CSV text, files, and creating lists

Part 1: Movie class (50 pts):

Start with the provided file **`movie.py`**. Add code to define a class **Movie.** This file **`movie.py`** should contain only a class definition; it should not include a **`main()`** function.

- The **Movie** class must contain ***at least*** the following **private** attributes:
  - `__title` (string)
  - `__release_year` (integer)
  - `__director` (string)
  - `__ratings` (list) *Hint: initialize to empty list.*
  - `__average_rating` (float) *Hint: initialize to 0.*
  - You can add more attributes if you like.
- The **Movie** class must contain ***at least*** the following methods:
  - **Private** constructor `__init__()` to create and initialize movie objects.
  - **Private** str method `__str__()` that returns a string describing a movie object for printing: "Title: _____ Year: _____ Director: _____"
  - **Public** method `add_rating()` to append a rating to a list of ratings. It is a VOID function; it does not return anything.
  - **Public** method `calc_average_rating()` that calculates the average rating and stores it in the `__average_rating` attribute. It is a VOID function; it does not return anything.
  - **Public** method `getRatings()` that returns the ratings list.
  - **Public** method `getAverageRating()` that returns the average rating.
  - Optionally you can add getters and setters for other attributes.

- The provided **`test_movie.py`** imports your **movie** module and its **Movie** class and tests all required methods. Do **not** edit **`test_movie.py`**. Put it into the same directory, and run it to test your **Movie** class. If your **Movie** class is correct, **`test_movie.py`** prints "All tests pass." When that works, proceed to Part 2.

Part 2: Movie Manager (50 pts)

Starting with the provided file movie_mgr.py, write a program that reads movie data from a CSV file and manages it. The program has two functions:

- **`read_movie_data(file)`**: takes a file that is already open for reading, reads in the movie data (1 line/movie), creates a movie-object from each data-line, stores all the movie-objects in a list, and returns the list.
- **`main()`**: allows users to search the movie data for specific movies, as follows:
  1. open provided movie_data file,
  2. call read_movie_data() with opened file,
  3. close file.
  4. Ask user for text to use to search the movie titles,
  5. Search list of movies for user-provided text,
  6. Print **all** movies for which the title includes the search-text.
     If no movies match the search-text, print "Movie not found."
  7. Ask user if wants to search for another movie,
     Validate response ("Y", "y", "N", "n"); re-ask (step 7) if necessary,
  8. If user responds "Y" or "y", return to step 4 (**not step 1**)
     If user responds "N" or "n", print "Goodbye!" and quit.

Submitting

Upload **`movie.py`**, **`movie_mgr.py`**, and **`readme.txt`** to **NT Lab 12** in Canvas. **Ask a TA, Tutor, or Professor for help if you need it.**