CS 101
Program 8
Algorithm due Thursday, Nov. 1
Program due Sunday, Nov. 8

For this program, we'll be writing a simple object (class) to simplify some basic data processing. We'll be using the movie-ratings file from earlier in the semester. To get full credit, **your program must be object-oriented as specified here.**

We're going to define a Movie object with certain properties and methods; your main program will import your Movie module and use the titles and ratings files to build up a database of movies. Your program will then report some basic statistics about your movie database (such as which movies were seen the most, which had the highest rating, which had the lowest, etc.). Finally, your program will save the database using the pickle module so the data can be read again in a single step, without having to build up the data from scratch again.

The Movie module
You will define a class called Movie. The definition will be stored in movie.py. Your class should have:

- An initializer (__init__). The initializer should have a parameter for the title, which defaults to the empty string. The initializer sets the title and all internal counters to 0. You will need counters for how many ratings the movie has received, and how many ratings of each value were received (Hated it, Didn't like it, So-so, Liked it, Loved it).
- Functions __str__ and __repr__. The __str__ method should return a string reporting the title, number of ratings, and average ratings. This should be formatted for user-friendly output. Sample output is shown below. The __repr__ method should return a string giving the title, and the counts of the individual ratings.
- A method called AddRating. This method takes a rating as a parameter. The only valid values for ratings are -5, -3, 1, 3, 5. Any other ratings that are passed in to this method (non-integers, or integers with other values) should be silently ignored (that is, it should not change any data, print any message, or cause any other problem with the program).
- AvgRating. This method returns the average rating based on the object's data. If the movie has no ratings, this method returns 0.0.

Main Program
Your program will read the files titles.txt and ratings.txt. These are the same files from program 5. Your program will start by using the titles.txt file to build up a list of movies. Then, for each line in the ratings file, assign each rating to the appropriate movie. Note that this ratings file is the original file from program 5, with a few invalid ratings. Your main program should pass them on to the objects without fixing them; the object will decide whether the rating is valid or not.

Your main program will then query the items in the list and report:

- The movie receiving the most ratings, with the number of ratings for that movie;
- The movie receiving the fewest ratings, with the number of ratings for that movie;
- The movie with the highest average rating, with the average; and
- The movie with the lowest average rating, with the average.
    - Display average values to 2 decimal places.

If there are any ties in the above data, list all tied values.

Finally, your program will save the movies list to a separate file, so you don’t have to process all the input files again the next time the program is run. Use the pickle module to save the list of movies into a binary output file called movies.dat. (We’ll go over how to do this in class.)

Development notes:

- Your program must use a Movie object, as specified. The movie.py file should contain the definition of the Movie class and nothing else. Your main program will import movie.
- Ratings data can either be stored in a list or in separate values.
- For calculating average ratings, “Hated it” has a weight of -5, “didn’t like” -3, “so-so” 1, “liked it” 3, “loved it” 5.
- Your main program handles interacting with the user and with the external files. It passes ratings data to each movie individually. Likewise, each movie reports its own number of ratings and average ratings.
- All functions, including object methods, must have docstrings.

```
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
The most seen film(s) in this class, with 68 reviews:
Avatar

The least seen film(s) in this class, with 12 reviews:
Dr Suess’ The Lorax
Madagascar 3: Europe’s Most Wanted


The best-reviewed film(s) in this class, with a rating of 4.18:
The Avengers (Marvel)

The worst-reviewed  film(s) in this class, with a rating of -0.89:
Twilight: Eclipse


Saving file....Done!
>>> 
```