COMP 30070 Practical Exam 2013

Complete part 1 first. After that, parts 2 to 4 can be attempted in any order? **Submit only one Ruby program finally**. Aim to get each part working correctly before moving on to the next. The bulk of the marks will be awarded for how much you get working, so be sure to submit a working solution, commenting out sections of code if necessary. Marks are also awarded for clear design and coding, indentation and (light) commenting. You may write your program in one .rb file if you wish (the simplest approach), or use several .rb files.

Turn off your wireless network and bluetooth before the exam commences.

Test cases: Full marks will be awarded for the following level of testing. For each class you write, write test cases that test one simple method in the class and one of the more complicated methods in the class.

Preamble

The round-by-round results of a 6-player, all-play-all (5-round) tournament are stored in a file `tournament.txt`; a sample is as follows:

```
Eimear 2000  1 1 1 0 1
Jack   1800   0 0 0 0 1
Ciara  1600  1 0 1 1 0
Jamie  1400  0 1 0 0 1
Luke   1200  1 0 0 1 0
Aoife  1000  0 1 1 1 0
```

The first string is the player's name. This is followed by the players rating, a positive integer value. The comes the round-by-round results. 1 indicates a win, 0 indicates a loss. Draws are not possible in this sport. The first line indicates that Eimear's rating is 2000 and she won her first three rounds, lost the fourth and won the fifth. Note that is not known who beat whom. This is just a sample input program; the program you write should be able to handle any number of players and any number of rounds.

1. A `Player` comprises the attributes as described above and has a `to_s` method that returns a string comprising the player's name, their rating, results and total score. Implement the `Player` class.

   Create a `Tournament` class to represent an arbitrary number of `Players`. Add a suitable `to_s` method to the `Tournament` class, and a method to compute the average rating of the players in the tournament. In your main script, create a `Tournament` object, load it with Players created with data from `tournament.txt`, and use the `to_s` method to print this `Tournament`. Using the sample data file above, the output of the program would be:

   ```
   Average rating is: 1500
   Eimear 2000  1 1 1 0 1 TOTAL: 4 points
   Jack   1800  0 0 0 0 1 TOTAL: 1 points
   ```

```
Ciara   1600   1 0 1 1 0 TOTAL: 3 points
Jamie   1400   0 1 0 0 1 TOTAL: 2 points
Luke    1200   1 0 0 1 0 TOTAL: 2 points
Aoife   1000   0 1 1 1 0 TOTAL: 3 points
```

(40 marks)

2. Add a `sort!` operation to the `Tournament` class that sorts the players into rank order of merit, from first place (most points) to last place (fewest points). Hence change the main program so that it outputs the players in this order.

(15 marks)

3. Add an operation to the `Tournament` class to check that the tournament data is *consistent*, i.e. that the number of rounds is one greater than the number of players and that the total number of wins in each round is the same as the total number of losses. In your main script, invoke this operation and inform the user whether or not any problems were found.

(20 marks)

4. The players' ratings are changed as a result of the tournament according to the following formula[1]:

$$R'_A = R_A + 40 * (S_A - N * E_A)$$

where R'_A is the new rating for player A, R_A is player A's original rating (given in `tournament.txt`), N is number of games player A played, S_A is player A's total score in the tournament and E_A is their expected score given by the following formula:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}.$$

where R_B is the average rating of player A's opponents. Extend your program to output the new rating for each player. Eimear's new rating should be 1966.

(25 marks)

Submission

Create a zip file called <student_id>.zip that includes the following:

1. All the .rb files you wrote, including one called `main.rb` that runs the main script for the program and a script `all_tests.rb` that runs all your tests.
2. A file called `statement.txt` that contains your name, student ID and a statement of what you achieved, e.g., "parts 1-3 completed correctly, part 4 attempted but unfinished"
3. A copy of the input file, `tournament.txt`.

[1] This is a simplification of the Elo rating system. Even if you're familiar with this system, just use the simplified formulae provided here.