CODING BLOX

Coding Blox is an Online Coding Platform that allows a user to Sign Up, Create Contests and participate in Contests hosted by Others.

- Each contest can have a level (LOW, MEDIUM, HIGH) and will contain a set of questions.
- Each question will have different levels of difficulty(LOW, MEDIUM, HIGH) and score.
- Based on the contest level, the question set is going to be decided. Contest level with LOW difficulty will have questions with LOW difficulty level.
- Final score will be decided based on the difficulty LEVEL chosen for a contest
- Users solve problems and get points based on the difficulty of the problems and after the contests scores of the users are updated.

You have to design the Coding Blox platform with the following functionalities.

Command : **CreateUser <user_name>** :

- Provided a user name register a user with a default score of 1500.
  Eg: CreateUser Ross
  CreateUser Monica
  CreateUser Joey
  CreateUser Chandler

Command: **CreateQuestion <difficulty_level> <score>**

- **Eg:** CreateQuestion “LOW” 10
- *AutoIncrement QuestionId should be assigned starting with 1.*

Command: **ListQuestion <difficulty_level>**

- ***difficulty_level*** *is optional if nothing is passed it’ll show all the questions. If difficulty_level is passed then list all the questions with that difficulty level.*
- **Eg: ListQuestion LOW or ListQuestion**

Command : **CreateContest <contest_name> <contest_level> <contest_creator_user_name>:**

- **Eg:** CreateContest “diwali_contest” LOW “Ross”.
- *It means Ross is creating a contest with name “diwali_contest”*
- *The contest_creator_user_name will always attend the contest.*
- *AutoIncrement ContestId should be assigned starting with 1.*

- *Question list is made independent of the contest and it is decided based on the contest difficulty level. For eg: If the contest difficulty level is LOW then all questions with LOW difficulty level can be used in the contest.*

Command : **ListContest <difficulty_level>:**

- ***difficulty_level is optional*** *if nothing is passed it'll show all the contests. If difficulty_level is passed then list all the contests with that difficulty level.*
- **Eg: ListContest LOW or ListContest**

Command: **AttendContest <contest_id> <user_name> :**

- *The contest_creator will attend the contest automatically.*
- **Eg:**
    - AttendContest 1 Monica
    - AttendContest 1 Joey

Command: **RunContest <contest_id> <contest_creator_user_name>:**

- **Eg:** RunContest 1 Ross
- *The user who has created the contest can only start the contest.*
- *Question list is made independent of the contest and it is decided based on the contest difficulty level.* ***For eg****: If the contest difficulty level is LOW then all questions with LOW difficulty level can be used in this contest.*
- *It should generate random questions solved by the users based on the contest difficulty* level chosen.
  - "Ross" : 1,3,5
  - "Monica" : 1,6,3
  - "Joey" : 2,4,6
- *Should update User Score based on the score secured according to the Problems solved. currentContestPoints = Sum of scores of all question solved*
- *Final Score should be calculated based on the difficulty level of contest*
    - For LOW level → newScore = currentScore + (currentContestPoints - 50)
    - For MEDIUM level → newScore = currentScore + (currentContestPoints - 30)
    - For HIGH level → newScore = currentScore + (currentContestPoints)

Command : **LeaderBoard <sorting order asc/desc>**

- *Should display a leaderboard with userIds and Score.*
- **Eg**: **LeaderBoard score desc**
  - "Joey" : 1515
  - "Ross" : 1485

- “Monica” : 1475

Bonus Functionality :

1. Implement a functionality where contest history can be displayed, given a contest id.
    - Command : **ContestHistory <contest_id>:**
        - *Should display a contest leaderboard with userName, points Secured and questions solved.*
        - **Eg: ContestHistory 1**
          - “Joey” : 65 [2,4,6]
          - “Ross” : 35 [1,3,5]
          - “Monica”: 25 [1,3,6]

2. Implement Contest Withdraw functionality where users can withdraw from a contest before the RunContest has executed.
    - Command: **WithdrawContest <contest_id> <username>**:
        - **Eg**: WithdrawContest 1 “Joey”
        - *The user who created the contest can’t withdraw from the contest.*

Expectations:

1. Create the sample data yourself. You can put it into a file, test case or main driver program itself.
2. Code should be demoable. Either by using a main driver program or test cases.
3. Code should be modular. Code should have basic OO design. Please do not jam in responsibilities of one class into another.
4. Code should be extensible. Wherever applicable, use interfaces and contracts between different methods. It should be easy to add/remove functionality without re-writing the entire codebase.
5. Code should handle edge cases properly and fail gracefully.
6. Code should be legible, readable and DRY.

Guidelines:

1. Please do not access internet for anything EXCEPT syntax
2. You are free to use the language of your choice
3. All work should be your own.