CS 213: Programming 2

Assignment 2 – Quiz game V2.0 - 6 Marks

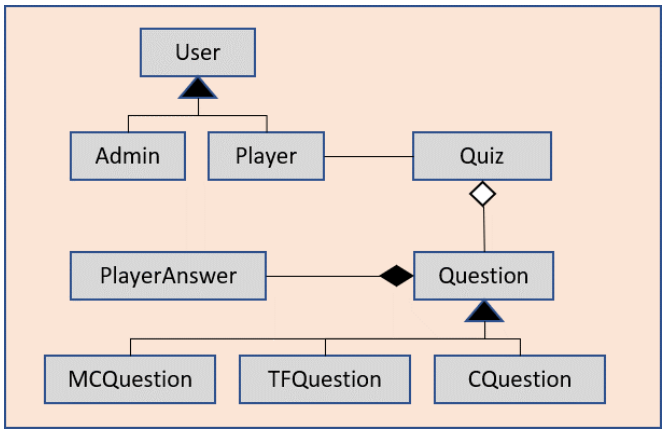
Prepared by: Sara Maadawy

Deadline: 24 November 2019 (Midnight)

1 QUIZ GAME INTRODUCTION

This assignment is an extension to assignment 1. You are required to design and implement Quiz game V2.0. The new version should support the following features and enhancements:

- V2.0 should support multiple users. This means that, while the program is executing, the user can choose to switch his account and enter the username of a different user. Each user will have a different history that will be available when the user is logged in.
- The new version will support two types of users: Admin and Player. The admin user will only be allowed to do administration tasks (i.e. access the administration menu) while the player will only be allowed to see his/her information as well as being able to start quizzes. The player will not have access to the administration menu unless he switches his account to an admin account. The admin will not be allowed to start a quiz unless he switches his account to a player account.
- A login page, where the user enters his username and password. Based on the entered details, the user role will be identified, and a related main menu will be displayed.
- A player 'main menu page' available for the 'Player' user.
- An admin 'main menu page' available for the 'Admin' user.
- Different types of questions should now be supported, new question types are:
    - MCQ Question: Same as in V1.0
    - True or False Question
    - Complete question
- While adding a new question or loading questions from file, you will need to check if the added questions are duplicated or not.
- The code should be implemented with OOP concepts. This means that,
    - At least, the following classes and their relationships are expected to be implemented in the delivered code (you can add more classes/relations if you need to). If you have another suggested design for the classes that you think will be easier or more efficient, you can implement it and explain the differences with your TA in your discussion.



- An inheritance relationship will be implemented between **Question** as super class and the other types of question classes as child subclasses. Same for the User class (superclass) and the subclasses Admin and Player (as shown in the figure above)
- Operator overloading will be used to check if the question is repeated or not by overloading the == operator to compare questions together. Same applies while adding a new user (you need to make sure that no users are added with the same username).

- The program execution supports multiple quizzes. This means that each user can take more than one quiz before they exit the program or switch the account.

2 QUIZ GAME FEATURES

The following features must be available in your program.

2.1 LOGIN PAGE

The login page (displayed below) is the main page that opens when the program executes. In the login page, the program supports the following activities:

- Asks the user for his username and password
- Searches for the username in the list of usernames in the program and checks if the username exists
- If the username is found, the entered password will be compared to the password associated with the found username
- If the passwords match, the role of the user will be identified, and the related menu will be displayed (i.e. player or admin menu). If the passwords did not match, the control will go back to the login page and a message will be displayed indicating that the user entered a 'Wrong username or password'.

```
**********************************************************************
                    Welcome to the Quiz game program V2.0!
**********************************************************************
Please enter your username: _
Please enter your password: _
```

2.2 ADMIN MAIN MENU PAGE

The admin main menu page will only be displayed when an 'admin' user logs in.

```
Welcome user user (ADMIN), please choose from the following options:
        [1] Switch accounts
        [2] Update your name
        [2] View all users
        [3] Add new user
        [4] View all questions
        [5] Add new question
        [6] Load questions from file
        [7] Exit the program
My choice: _
```

2.3 PLAYER MAIN PAGE MENU

The player main menu page will only be displayed when a 'player' user logs in.

```
Welcome user user (PLAYER), please choose from the following options:
        [1] Switch accounts
        [3] Update your name
        [4] Start a new quiz
        [5] Display your scores statistics
        [6] Display all your scores
        [7] Display details of your last 2 quizzes
        [8] Exit the program
My choice: _
```

2.4 PLAYER AND ADMINISTRATION MENU: SWITCH ACCOUNTS

The user should be able to logout by selecting the 'switch account' option. That option will return the user to the main login page waiting for the user to enter a username and a password for an existing user in the program.

2.5 ADMINISTRATION MENU: VIEW ALL USERS

The admin should be able to view all existing users. A table will be displayed to the admin with all first names, last names, username and roles of existing users. A suggested view is illustrated below.

```
Existing users in the system:
First name      Last name       Username        Role
User            user            admin           admin
User            user            player          player
John            smith           admin           admin
John            doe             player          player
Press [n] if you want to add a new user or [b] to go back to the main menu.
```

2.6 ADMINISTRATION MENU: ADD NEW USER

The admin should be able to add new user. Use '==' operator overloading to check if the admin entered a duplicate username or not. The admin should not be allowed to add a user with an already existing username. While adding a new user, the admin should add the user's first name, last name, username, password and role. After the admin adds the new user, the new user can immediately login using the 'switch accounts' feature.

2.7 ADMINISTRATION MENU: ADD NEW QUESTION

The admin user, through the administration menu, can add as many questions as he needs. The program supports three questions type:

- **MC question:** it supports 4 choices with only one correct answer. So, multiple correct answers are not supported. While the admin adds the question, he should specify which choice is the right choice.
- **True or False question:** The admin will add a statement and will specify of this statement is true (T) or false (F).
- **Complete question:** The admin will add a statement with one word replaced by dots, he will also specify the correct answer. The question supports only one word as an answer and one support (i.e. multiple answers for the same spot are not supported as well as multiple spots per statement), for example
    - **Supported (one answer – one spot):** *At …. German declared war on Russia and France.* (answer: 1914)
    - **Not supported (One answer per spot – two spots):** *At 1914 German declared war on … and …* (answer: Russia, France)
    - **Not supported (Multiple answers per spot):** *At 1914 German declared war on …* (answer: Russia or France)

Any weights of the questions should be: 3 marks for a complete question, 2 marks for a MC Question and 1 mark for a TF questions. Any added question regardless of its type should have a unique question ID which is automatically generated (i.e. 1,2,3,..)

2.8 ADMINISTRATION MENU: VIEW ALL QUESTIONS

The admin user, through the administration menu, should be able to view all existing questions and delete existing ones from the display list as shown in the figure below. That screen should also show the

```
Number of questions available: 6
---------------------------------------------------
MC Questions list (Total: 2 Questions)
---------------------------------------------------
[1] (ID: 1) Which one of the following is a flightless bird
        [a] swan        [c] duck        [d]hen          *[e]emu
[2] (ID: 5) Which is the capital of Egypt?
        [a] Alexandria  [c] Oman        *[d]Cairo       [e]Luxor
---------------------------------------------------
TF Questions list (Total: 2 Questions)
---------------------------------------------------
[1] (ID: 2) A swan is a flightless bird (Answer: F)
[2] (ID: 3) An emu is a flightless bird (Answer: T)
-----------------------------------------------------------
Complete Questions list (Total: 2 Questions)
-----------------------------------------------------------
[1] (ID: 7) Hundred-year war was fought between … and England. (Answer: France)
[2] (ID: 9) The branch of science the studies cells is called …. . (Answer: cytology)

-------------------------------------------------------------------------------
Press [d] and the question ID if you want to delete a question (Example: d 2)
Press [b] if you want to go back to the main menu
```

total number of questions in the program as well as the total number of each type. The right answer for each question should be highlighted/displayed. If the user selected to delete a question, the list should be displayed again without the deleted question. The questions should be displayed organized by the question type. The ID of each question should be displayed so that the admin can use it to delete the question if he needs to.

2.9 ADMINISTRATION MENU: DELETE A QUESTION

That feature will be available from within the 'view all questions' screen. The user will type the question ID for the question they would like to delete. After deleting the question, the list of questions will be displayed again without the deleted question. The ID will not be given to any other question in the future until the end of the program execution.

2.10 Administration menu: Load questions from file

The user should be able to enter a file name that contains the questions and the program will load it. The file should have the following structure:

- A line will be dedicated for the question type. It will contain one of three values (MCQ, TF or COMPLETE). If any other text is added other than these three values, an error will be displayed to the user and the process of loading the file will stop (only the questions up to this error will be loaded).
- A line will be dedicated for the question body, that will apply to all question types.
- In case of MC questions, the next four lines will be dedicated to the 4 choices of the question with the first line being the correct choice.
- In case of true or false questions, the next line will contain either the word 'true' or 'false' to indicate the answer of the question. Any other content will result in an error message, the load operation will not continue and the question loading process will only load up until the previous question (since this one contained an error)
- In case of a complete questions, the next line will contain the correct answer to the question. An empty content will result in an error message, the load operation will not continue and the question loading process will only load up until the previous question (since this one contained an error).

```
MCQ
<q1_text>
<q1_correct_choice>
<q1_choice2>
<q1_choice3>
<q1_choice4>
TF
<q2_text>
<true or false>
COMPLETE
<q3_text>
<answer text>
<q4_type>
….
```

Note that the questions in the file will not have a specific order. This means that questions can be of any type and doesn't have to exist in groups of the same question type. That is why, each question must be preceeded with its type.

The class 'Question' must include a method 'readQuestionFromFile' which will be overrided on each question type and will be read in each question according to the question structure.

A sample of a file that can be loaded to test your program is attached to this assignment.

2.11 Player and Administration menus: Update your name

By default, the program will have two users / usernames when it first starts: 'PLAYER' with a player role and 'ADMIN' with an admin role. Each will have 'user' as a first name and 'user' as last name by default. If the user would like to update his first/last names, they should be allowed to do that. Note that a username should not be updated. Only the first and last names. The username is used for login while the first and last names are mainly used to communicate with the user (i.e. in the main menu for example).

2.12 Player menu: Start a new quiz

The quiz should include a specific number of questions that is constant throughout all the quizzes (for example: 5 questions/quiz). The quiz should be randomly generated from the question pool. If the number of questions in the pool is less than the number of questions required by the quiz, then an error message should be displayed for the user enquiring them to add more questions. Like the questions, the choices for each question should be randomly displayed (i.e. different random order each time).

The number of question types in a quiz must be fixed so that all quizzes will finally have the same grade (all types must be included in one quiz). So, for example, if the program is designed to have five questions, then all quizzes should (for example) have 1 complete question, 3 MC question and 2 TF question so that all quizzes will always have a grade of 10 (1*3 + 2*2 + 3*1). You are not restrained by a specific grade; you just need to have all quizzes have the same total grade whatever that is. That will help when you calculate the averages for the user.

After the user finishes the quiz, a report will be displayed with his score, number of right answers and number of wrong answers. The control will then go back to the player main menu.

2.13 Player menu: Display score statistics for the player:

The program allows the users to view their score statistics after any quiz if they selected the option to. The score statistics should at least include the highest and lowest scores the user achieved during the execution as well as the average score for the number of quizzes they took on that execution. The number of exams the user took should also be listed (See figure below).

```
Your score statistics per quiz:
        - Number of Quizzes taken: 5
        - Highest quiz score: 9/10
        - Lowest quiz score: 5/10
        - Average quiz score: 6.8/10
Your score statistics per question type:
        - Number of MC questions: 10
        - Number of Complete questions: 5
        - Number of T/F Questions: 15
        - Average grade for MC questions: 0.8/2
        - Average grade for Complete questions: 1.8/3
        - Average grade for T/F questions: 0.9/1
Press [b] if you want to go back to the main menu or [e] to exit
My choice: _
```

2.14 Player menu: Display all scores

The program will allow the user to display a history of the quizzes he finished during the program execution session. This feature should allow the user to display a detailed report of his previous quizzes that includes: The total number of quizzes he finished and for each quiz, the number of right and wrong answer and the score for each individual quiz.

2.15 Player menu: Display details for your last 2 quizzes

The program will allow the user to explore his answers in the last 2 quizzes he took. That includes for each quiz:

- displaying all the questions in the quiz that the user took,
- the answers the user entered and if the answer was right or wrong
- the right answer for the question
- the user's score for that question
- the total score for that quiz

3 General notes and Considerations

Please consider the following while doing your assignment:

- You are free to update your existing code or do the assignment from scratch. The TAs will release a model-answer code for assignment-1 that you can also use if you like.
- You should form a team of 3-4 students, all students must contribute in the code.
- Submit your compressed code and rename it using the following format <Asgmnt2_G#_ID1_ID2_ID3_ID4>.<zip/rar>
- All students must understand the code, even the parts they did not write.
- **Error handling:** Consider the wrong input that the user can put and plan for it. Display error/warning/information messages as needed when that happens and make sure that the wrong input from the user won't break your program.
- **Standard conventions:** Follow standard code conventions to produce a clean and well document code. A good rules-guide to help you achieve this is in this link.
- **Do not share your code outside your group before your discussion.**