CS 213: Programming 2

Assignment 1 – Quiz game: 6 Marks

Prepared by: Sara Maadawy

# Quiz game introduction

In the quiz game, you are required to design and implement a single-user quiz game. The game should include the following features:

* A main menu with the options available for the user. The options are displayed in Figure 1.

Welcome USER, please choose from the following options:

[1] Go to administration menu

[2] Update your name

[3] Start a new quiz

[4] Display your scores statistics

[5] Display all your scores

[6] Exit

My choice: \_

Figure 1 Main menu of the Quiz Game

* The program execution supports a single user and multiple quizzes. This means that the user can take more than one quiz before they exit the program.
* The program allows an administration menu where the user can add more questions to the questions pool and delete existing questions as well.

Figure Administration menu

Welcome to the administration menu, please choose from the following options:

[1] View all questions

[2] Add new question

[3] Load questions from file

[4] Go back to main menu

My choice: \_

# Quiz Game Features

The following features must be available in your program.

## Administration menu: Add new question

The user, through the administration menu, can add as many questions as he needs. The only supported question type in this version of the program is MCQ with 4 choices and only one correct answer. So, multiple correct answers are not supported. Also, all questions will have the same grade.

The Question data should include: Question ID (Should be unique and automatically generated, i.e. 1,2,3,…), Choice1, Choice2, Choice3,Choice4. It is up to the student to choose how to represent the correct answer of each question.

## Administration menu: View all questions

The 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 in the program. The right answer for each question should be highlighted. If the user selected to delete a question, the list should be displayed again without the deleted question.

Figure A suggestion of how the 'View all questions' screen can look like

Number of questions available: 2

Questions list:

-------------------

[1] Which one of the following is a flightless bird

[a] swan [c] duck [d]hen \*[e]emu

[2] Which is the capital of Egypt?

[a] Alexandria [c] Oman \*[d]Cairo [e]Luxor

-------------------------------------------------------------------------------------

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

## 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.

## 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:

<q1\_text>

<q1\_correct\_choice>

<q1\_choice2>

<q1\_choice3>

<q1\_choice4>

<q2\_text>

<q2\_correct\_choice>

….

A [sample of a file](exam_questions.txt) that can be loaded to test your program is attached to this assignment.

## User menu: Update your name

By default, the program will assume the user name is ‘USER’ and will use it to communicate with the user. If the user would like to update this, they should be allowed to enter a new name that will be used instead.

## User 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).

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 main menu.

## User menu: Display score statistics for the user:

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:

- Number of Quizzes taken: 5

- Highest score: 9/10

- Lowest score: 5/10

- Average score: 6.8/10

Press [b] if you want to go back to the main menu or [e] to exit

My choice: \_

## User 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.

# General notes and Considerations

Please consider the following while doing your assignment:

* **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 this won’t break your program.
* **Standard convention:** 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](https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29).
* **Do not share your code outside your group before your discussion.**