**Title: Open ended lab**

**Introduction**

The "Project" is a comprehensive Java program designed as a semester project. It serves as an interactive quiz application, testing the user's knowledge in English, Math’s, Physics, and Chemistry. This project not only assesses the user's understanding of various subjects but also incorporates essential programming concepts, such as file handling, user authentication, and object-oriented design.

**Classes**

#### 1. Questions Class

- \*\*Attributes:\*\*

- `questionText`: An array storing the text of the questions.

- `questionAns`: An array storing the correct answers to the questions.

- `count`: An integer counting the number of correct answers.

- `wrong`: An integer counting the number of incorrect answers.

- \*\*Methods:\*\*

- `Questions(String[] text, char[] ans)`: Constructor for initializing questions and answers.

- `getQuestionText()`: Returns the array of question texts.

- `getQuestionAns()`: Returns the array of correct answers.

- `getCount()`: Returns the count of correct answers.

- `getWrong()`: Returns the count of incorrect answers.

- `resetCount()`: Resets the count and wrong counters.

- `displayQuestions()`: Displays questions, accepts user input, and evaluates correctness.

- `isValidOption(char opt)`: Checks if the user input is a valid option.

- `isCorrectAnswer(char opt, int questionIndex)`: Checks if the user's answer is correct.

#### 2. Project Class

- \*\*Attributes:\*\*

- `englishQuestions`, `mathsQuestions`, `physicsQuestions`, `chemistryQuestions`: Instances of the Questions class for different subjects.

- \*\*Methods:\*\*

- `main(String[] args)`: Entry point of the program.

- `menu()`: Displays the menu with test options.

- `loadingAnimation()`: Displays a loading animation.

- `initializeQuestions(String subject, Questions questions, Scanner scanner)`: Initializes and displays questions for a specific subject.

- `getTotalScore(Questions... subjects)`: Calculates the total score across multiple subjects.

- `getTotalWrong(Questions... subjects)`: Calculates the total number of incorrect answers across multiple subjects.

**Program Flow**

1. \*\*User Authentication:\*\*

- Asks for the user's name.

- Checks if a file named "a.txt" exists.

- If not, asks for a password and proceeds to the quiz.

2. \*\*Quiz Structure:\*\*

- Presents a menu of four subjects: English, Maths, Physics, and Chemistry.

- For each subject:

- Displays questions.

- Accepts user input and evaluates correctness.

- Records scores and wrong answers.

3. \*\*Result Presentation:\*\*

- Displays the total marks (20) and the user's test score.

- Displays the total number of incorrect answers.

4. \*\*Result Recording:\*\*

- Appends user information, test score, and wrong answers to the "a.txt" file.

**Conclusion**

The Project Java program offers an engaging and educational quiz experience. It not only assesses the user's knowledge but also demonstrates the application of Java programming concepts. The incorporation of user authentication, dynamic question handling, and result recording showcases a well-rounded understanding of software development. This project serves as a testament to the student's proficiency in Java and their ability to create interactive and meaningful applications.